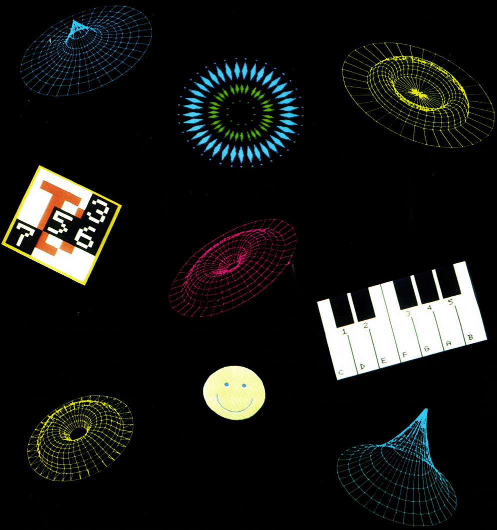
MSX
たのしいゲームの作り方

日本ナレッジインダストリ株式会社編著

西東社

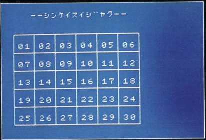
MSX
たのしいゲームの作り方

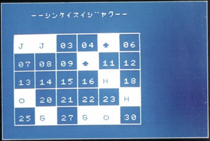
日本ナレッジインダストリ株式会社編著



西東社

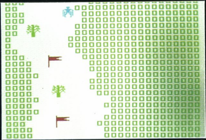
神経衰弱

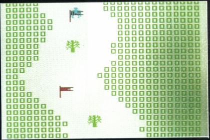




ズラリ並んだ30枚のカード。どれとどれが合うかは、カンと記憶の勝負だ。何人かでやるか、パソコン相手に一人で挑戦するか、敵の神経を衰弱させてしまおう。

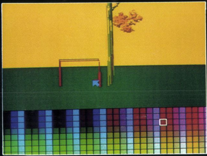
スキーゲーム

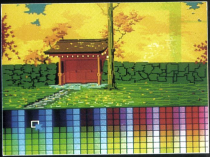




画面は雪のゲレンデ、国体コース。めまぐるしく変化するコースを滑走し、赤い旗を取れば、旗門通過で1点だ。トップレーサーをめざすキミ、何点までいけるかな?

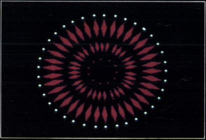
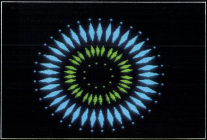
お絵かきソフト





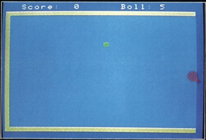
下に並んだカラフルな256色。これがパレットで、青い矢印が絵筆。直線、曲線はもちろん、色づかいしだいでは、ボカシもOKだ。あざやかCGのできあがり。

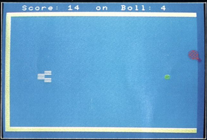
花火大会

夏の夜空の打ちあげ花火。赤、青、黄と、カラフルに咲いては消える火のうたげ。しばしの間、じっと目をこらそう。そっと、耳を傾けよう。

スカッシュを決めろ

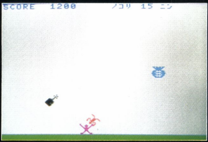




壁打ちテニスとでもいうべきスカッシュ。三方からはね返ってくるボールを、ラケットで打ち返そう。得点がアップすると、ボールが消えたり、障害物があらわれるぞ。

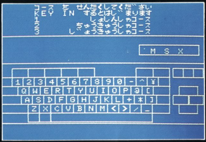
おたすけゲーム

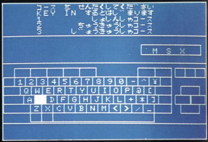




悪の宇宙帝国軍団から、捕虜と盗まれた大金が送り返されてくる。ただし、どこから落ちてくるのかわからないし、危険な爆弾も落ちてくる。正確にキャッチしよう。

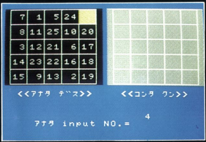
キーボード練習

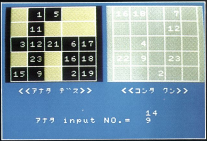




キーボードを使った"モグラたたき"。画面のどこかにあらわれてはすぐ消えるキーを、すばやく押していこう。知らないうちに、キー操作がうまくなるぞ。

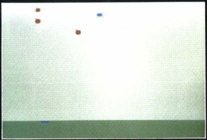
ザ・ビンゴ

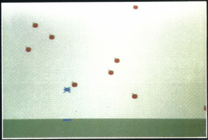




パーティーなどで人気のビンゴ。パソコンとキミで数字をいい合って、早くどこか1列が開いたほうの勝ち。コンタ君のパネルがそろわないように、数字を選べ。

着陸ゲーム

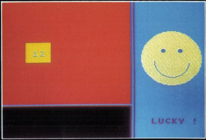




人類は新しい星に移住できるか、すべてはキミの宇宙船操縦テクニックにかかっている。雨のように降ってくるいん石をさけて、指定ポイントに無事、着陸させよう。

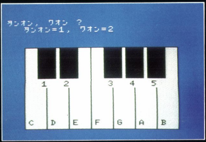
くろひげ恋うらない

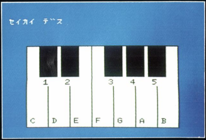




25枚のパネル。ここにキミの恋の行方が隠されている。数字を選んで、1枚ずつ開いていこう。ドクロがあらわれてジ・エンドか、くろひげがほほえんでくれるか……。

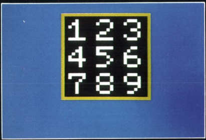
音当てクイズ

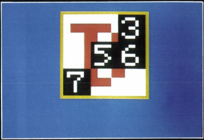




パソコン版音感テスト。単音か和音を選んで、鳴らされた音を当てる。自信のない人は、さけたほうが無難かも。得点が低いと、「ヤーイ　オンチ」とバカにされるぞ。

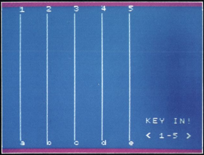
隠されたアルファベット

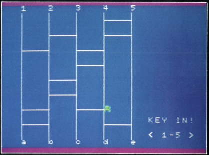




9枚の黒いパネルの下に、アルファベットが隠されている。1枚ずつ開いていき、何枚めで当たるか。キミの運命を決める文字があらわれるかも。

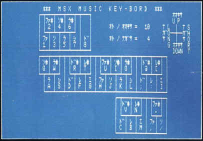
アミダクジ

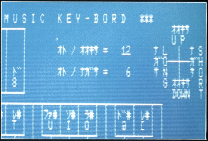




単純なんだが、意外におもしろいアミダクジ。キミが数字を選ぶと、アミダ君がテーマ曲に乗って進んでいく。きょうはツイているか、このクジで試してみよう。

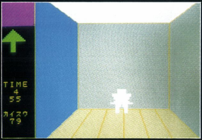
ミュージシャンはキミ!

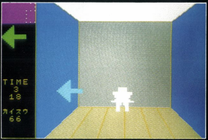




画面に鍵盤が並ぶと、キミのキーボードはミュージック用に早変わり、音の大きさ、長さを自由に決めて、3オクターブ、3重和音のパソコンサウンドを楽しもう。

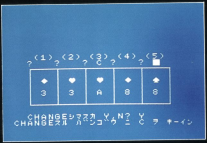
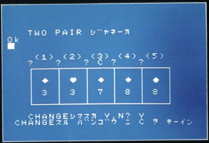
迷路の館





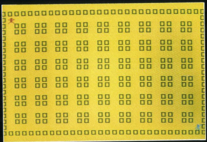
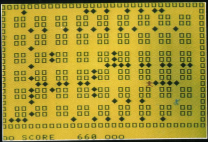
だれ一人として生きて出られた者はいないという、恐怖の迷宮。左上に表示される現在地を参考に、前後左右と進んで、みごと脱出をめざせ!

ポーカーゲーム

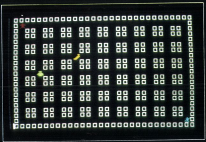
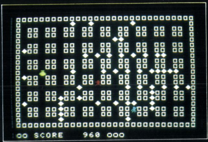
カードギャンブルの代表格ともいえるポーカー。どれをチェンジしてどの役をねらうかは、キミの腕しだい。パソコン相手に、どこまでポーカーフェイスが通じるか！

逃亡ゲーム

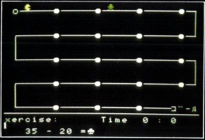
逃げても逃げても追ってくるインベーダ。キミの武器（？）はバリケードだけだが、それも食われてしまう。どれだけバリケードを築いて逃げられるか――逃亡の旅が続く。

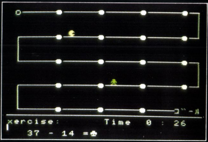
逃亡ゲームPARTII

逃亡ゲームの改造版。バナナとメロンを取ると、得点は大幅にアップする。ところが、それを敵に食われると……。このゲームではある得点までいくと、キミの勝ちになる。

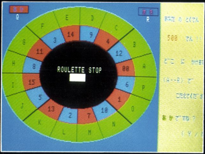
すごろくゲーム

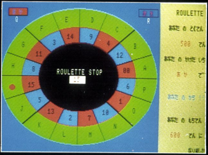




数字に弱い人にはご遠慮願いたい、すごろく計算ゲーム。すばやく答えを出すと先へ進めるが、まごまごしているとパクパクマンが追ってくる。キミは何秒でゴールできるかな?

カジノでルーレット

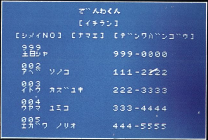




色を選ぶか、ズバリ数字を当てるか、カンとともに、キミの勝負度胸が試されるルーレット。盤の上をグルグルまわる球にキミの運命をかけて、億万長者を夢見よう。

電話ファイル





現代のコミュニケーションに欠かせない電話だが、電話帳ではチト古い。パソコン電話ファイルなら、変更も一覧表示もお手のもの。キミの交遊範囲も広がるぞ。

●はじめに

パソコンをはじめ、コンピュータ・ゲームに時間を忘れて熱中してしまった、という経験は、多くの人がお持ちでしょう。コンピュータを使ったゲームは、これまでのゲームにはないおもしろさと、奥行きの深さを持っています。

市販のゲームで遊びながら、自分にもゲームがつくれたら、と思った人も大勢いるのではないでしょうか。専用の言語を少し学べば、自分のつくったオリジナルのゲームで友人を遊ばせて優越感にひたる、なんてことも夢ではなくなります。

本書は、MSX パソコンを使って、ベーシックという言語によるゲームと、プログラムのつくり方を解説したものです。MSX パソコンは、どのメーカーのものでもプログラムや機器に互換性があり、しかもゲームをつくるうえでの機能、とくにグラフィックスにすぐれた特徴を持っています。

ゲームのプログラムをつくるのは、家を建てるのによく似ています。家を建てるには、まず設計図を書きます。同じようにゲームでも、内容はどうするか、どのような画面を使うか、終わり方はどうするか、といったアイデアをデザインして設計図のようなものをつくります。

設計図をもとに、実際に家を建てていく作業が、ゲームではプログラミングになります。プログラミングは、たんなるデザインとは違い、見ようみまねではできません。言語を理解し、その使い方を覚えていかなければ、コンピュータはけっしていうことを聞いてくれないのです。

このような点をふまえ、本書ではゲームだけで遊ぶことが目的の人と、プログラムづくりを学んでみたい人の、両方を対象に意図しました。

本書の読み方を説明しておきましょう。まず、プログラムを打ちこんだら、実際にゲームを楽しんでみてください。ゲームのおおよその内容がわかれば、次にプログラム解説を参考に、プログラムを1行ずつ解読していきます。最初はむずかしいものですが、いくつかのプログラムを読んでいくうちに、画面をどのようにつくるか、スコアはどのようにかぞえるのか、といったことがおぼろげにでもわかってくるはずです。

プログラムをひととおり解読したら、今度は自分で何か所かをいじってみて、ゲームを改造してみましょう。たとえば変数の値を少し変えただけでも、ゲームが違ったものになります。途中でエラーが出ることもあるでしょうが、そのときは、エラーの原因を徹底的につきとめてください。こうした作業をくり返すうちに、プログラミングの力が自然とついてくるものです。

各ゲームは、原則として行数の少ないものから順に並べました。といっても、順番どおりにやる必要はまったくありません。興味のあるゲームから、どんどん打ちこんで実行してみてください。

入力したプログラム、改造したプログラムは、必ずカセットテープかフロッピィディスクに保存してください。方法は、最初の「プログラム入力の基本と注意」の項で説明してあります。

なお、ゲームのタイトルの横に「MSX 2 対応」と明記してあるものは、MSX パソコンではできません。MSX 2 パソコン、あるいはMSX 2 用のアダプタを用意してください。

本書をもとに、市販の人気ソフトを上まわるゲームが生まれることを期待しています。

編　者

もくじ

●口絵／本文写真＝平野時義
●イラスト＝西山幸男
●協力＝株式会社東芝（本書のプログラム・リストは、
東芝の熱転写プリンタHX-P560で打ち出したものです）

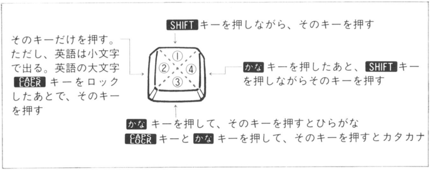
プログラム入力の基本と注意

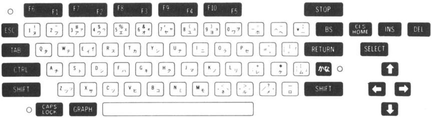
キー、文字の種類と入力の方法

MSX パソコンのキーボードは、ほぼ下図のようになっています(カナが50音順に並んだ機種もある)。キーは、色によって 2 種類に大別されていて、一方が文字キー、他方が特殊キーと呼ばれます。プログラムの入力に使うのは、このうち文字キーと、一部の特殊キーです。

文字キーには、一つのキーに 2 ~ 4 種類の文字や記号が描かれていて、上下、左右の 4 か所に分けることができます。これらをそれぞれ画面に表示するには、SHIFT かな CAPS LOCK の三つのキーを組み合わせて使い分けます。このうち、かな キーと CAPS LOCK キーは、一度押すと少しへこんだ状態になるかランプがついて、もう一度押すまで元に戻りません。これを、ロックするといいます。

本書の画面やプログラムの文字は、一部を除いてほとんどが英大文字とカタカナになっていますが、プログラムの実行上は、英小文字やひらがなでもさしつか





えありません（原則として〝〟ではさまれた文字や記号は、そのとおり入力する）。

ただし、LIST指令でプログラムを確認するとき（後述）、プログラムは英大文字で表示されますから、本書のように入力していったほうがよいでしょう。また、ゲームによっては、指示を英大文字でしか受けつけないものもありますから、使うときは、いつも CAPS LOCK キーをロックしておいてください。もちろん、ひらがなや英小文字を入力するときは、ロックを解除する必要があります。

●――特殊キーの種類と働き

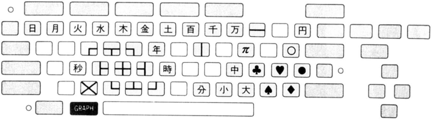
前述した3種のキー以外で、プログラムの入力やゲームの実行でよく使うものについて、次ページにまとめておきます。このうち、GRAPH キーと F1/F6 ～ F5/F10 （ファンクション・キー）については後述します。

●――グラフィック・シンボルの入力

プログラム中、キーに描かれていない記号や漢字、けい線などは、グラフィック・シンボルと呼ばれます。これらを入力するには、スペース・キーの左にある GRAPH キーを押しながら、該当するキーを押します（CAPS LOCK キーをロックしたままでもよい）。

なお、機種によっては、グラフィック・シンボルがキーの側面に描かれているものもあります。それがない場合は、付属のシールがついているはずですから、キーボードにはっておくとよいでしょう。

本書のプログラムでは、グラフィック・シンボルを緑色であらわしています。



●――プログラムの入力

プログラムとは、パソコンにある仕事や処理をさせるために、命令を順に並べたものです。各1行は、頭につけられた行番号と、命令文から成り立っています。

プログラムを入力するときは、次の手順と注意をしっかり守って行ってくださ

◉ 主な特殊キーの種類と機能

キー	名称	機能
RETURN	リターン・キー	プログラムの１行ぶんを入力し終えたときに、このキーを押してコンピュータに入力終了を知らせる
TAB	タブ・キー	押すと、カーソルが８文字単位で右へ移動する。その間は空白になる
CTRL	コントロール・キー	一般キーと組み合わせて使い、特殊キーと同じような機能を行う
STOP	ストップ・キー	①押すと、プログラムや命令の実行を一時停止する。もう一度押すと、実行が再開する。一時停止中は、キーによる入力はできない ② CTRL を押しながらこのキーを押すと、実行は中断し、「Ok」表示とカーソルが出る。この場合は、キーによる入力ができる
CLS HOME	ホームクリア・キー	①押すと、カーソルが画面の左上に移動する ② SHIFT キーを押しながらこのキーを押すと、カーソルが画面左上端に移動すると同時に、画面上の表示がすべて消える
← → ↑ ↓	カーソル移動キー	カーソルを矢印の方向へ移動する
INS	インサート・キー	押すと、カーソルの形が変わり、この状態でキーを押すと、文字がカーソルの左に挿入される。この状態を解除するには、このキー、あるいはリターン・キーやカーソル移動キーを押す
DEL	デリート・キー	押すと、カーソルの位置の文字が消え、カーソルの右側にある文字が左へ１文字ぶん移動する
BS	バックスペース・キー	押すと、カーソルの左側の１文字が消え、カーソルの位置にある文字が左へ１文字ぶん移動する
	スペース・キー	押すと、１文字ぶんの空白ができる

い。せっかく入力したのに、実行するとうまくいかない——という原因のほとんどは、入力ミスつまりキーの押しまちがいによるものです。

①画面に前のプログラムの文字などが残っている場合は、SHIFT キーを押しながら CLS/HOME キーを押すと、すべて消去される。ただしこの方法は、パソコンの記憶まで消してしまうことはない。

②新しいプログラムを入力するときは、NEW(ニュー) RETURN をキーインする。これで、パソコンに記憶されているプログラムやデータが消去される。

③行の先頭には必ず行番号をつけて、1行ずつ入力する。本書のプログラムを入力する場合は、それに従ってキーインしていく。行番号と命令の間は、必ず1字ぶんあけること。

④1行入力したら、最後に必ず RETURN キーを押す。本書のプログラム中、一つの行番号で2行以上になっている場合もあるが、行番号が変わるまでは1行として、続けて入力すること。

⑤入力が終わったら、LIST(リスト)(または F4/F9) RETURN で打ちこんだプログラムを画面に出し、ミスがないかどうかを調べる。

●——プログラム入力のポイント

プログラムは、あとで誤りをチェックしたり、働きを調べるために、見やすい形で入力します。そのポイントを、例とともにあげておきましょう。

①行番号のあとにアポストロフィ(´)がついた行は、注釈文。例のように、タイトルをつけたり、プログラムの中である処理を行うブロックの先頭につけて、プログラムをわかりやすくする。

注釈文は、プログラムの実行にはまったく関係がないので、入力しなくてもよいし、アポストロフィだけでもよい。

```
      ┌─①
      ↓            ②
99   'MSXゲーム
100 SCREEN□2:COLOR□15,4,7:CLS
105 OPEN "GRP:" AS #1
110 FOR I=0 TO 6
120 □LINE (I*24+24,48)-(I*24+24,168),15
130 NEXT I
140 PSET (45,20),4:COLOR 15,4:PRINT #1,"MSX□ノ□オタノシミ"
150 FOR J=1 TO 2000:NEXT J
   ③                                              ④
```

②各命令のあとは、1字ぶんあけると、プログラムが見やすくなる。
③例で行110～130の FOR（フォー） ～ NEXT（ネクスト） は、その間にはさまれた処理命令を、指定の回数だけくり返す命令のこと。はさまれた命令（例では行120）などは、1～2字ぶんさげて書いておくと、あとでチェックするのに便利だ。
④ " " ではさまれた文などは、多くの場合そのとおり画面に表示されるもので、原則としてそのまま入力すること。たとえば、例の④の空白は、詰めて入力すると、そのように画面に表示される。

●——ファンクション・キーで効率よく入力

キーボードの最上列に並ぶ F1/F6 ～ F5/F10 の五つのキーは、ファンクション・キーと呼ばれ、あらかじめある機能がセットされているものです。セットの内容は別表のとおりで、そのまま押すと F1 ～ F5 、SHIFT キーを押しながらそのキーを押すと、F6 ～ F10 の働きをします。以下の解説で F1 ～ F10 とあるときは、このようにキーを操作してください。

このうち、プログラムの入力と実行でよく使われるものを、簡単に説明しておきましょう。

① F2 ——行番号を自動発生させる

押すと、画面には「auto」と表示され、1字ぶんの空白のあとにカーソルが表示される。本書のプログラムのように、行100から始まって、10刻みで行番号をつ

◉ファンクション・キーのセット内容（␣は1字ぶんの空白）

キー	セット内容	機能
F1	color␣	色を指定する
F2	auto␣	行番号の自動発生
F3	goto␣	指定の行へ飛ぶ
F4	list␣	プログラム・リストを表示
F5	run RETURN	プログラムの実行
F6	color 15, 4, 7 RETURN	画面の色を初期状態にする
F7	cload"	カセットテープのプログラムを読みこむ
F8	cont RETURN	プログラムの実行を継続する
F9	list. RETURN ↑ ↑	プログラムの特定の1行を表示して、カーソルをその行に戻す
F10	CLS run RETURN	画面の表示を消してから、プログラムを実行する

けるには、まず行99の注釈文を入力してから、次のようにキーを押せばよい。

F2 (auto) 100, 10 RETURN

行番号の自動発生を解除するには、CTRL キーを押しながら STOP キーを押す。

②F4 ——プログラム・リストを表示する

プログラムの入力が終わったときや、誤りの訂正のとき、打ちこんだプログラムを画面に表示する。押すと、画面には「list」と表示され、1字ぶんの空白のあとにカーソルが表示される。このあとに続く数字のつけ方によって、次の5種類の方法がある。

F4 (list) RETURN →プログラムのすべてを表示

F4 (list) 100−500 RETURN →行100〜500を表示

F4 (list) 300 RETURN →行300を表示

F4 (list) −500 RETURN →最初から行500までを表示

F4 (list) 300− RETURN →行300から最後までを表示

③F5 、F10 ——プログラムを実行する

プログラムの実行と停止の項を参照。

④ファンクション・キーの設定

プログラムの入力でよく使う命令などは、自分で設定しておくと、入力時間が短縮できる。たとえば、F1 に「SCREEN」という命令を設定するには、次のようにキーを押す。

KEY 1, "SCREEN" RETURN

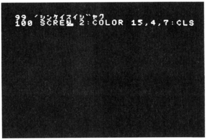
こうすれば、あとは F1 を一つ押すだけで、画面に「SCREEN」と表示される。なお、通常の画面では、最下列に F1 〜 F5 にセットされている内容が表示され、SHIFT キーを押すと、F6 〜 F10 の内容が表示される。

●——プログラムの修正、削除

プログラムの修正や削除は、1字だけの場合もあれば、数行単位で行うこともあります。くわしくは、付属の操作マニュアルなどを見ていただくとして、ここでは、操作のポイントだけをあげておきます。

注意するのは、プログラムの修正などをしたとき、その行のどこかにカーソルを合わせてから RETURN キーを必ず押すことです。プログラムの入力と同じく、このキーを押すまでは、ただ画面で修正されているだけです。RETURN キーを押してはじめて、命令などの内容がパソコンの頭脳に伝わるわけです。

①打ちまちがった文字の上にカーソルを合わせ、正しい文字を入力する。
②カーソルがミスした文字の一つ右側にあれば **BS** キーを押す。カーソルの一つ左の文字が消え、そのぶんが詰まる。
③よぶんに文字を打ってしまったときは、その文字にカーソルを合わせ、 **DEL** キーを押す。その文字が消え、そのぶんが詰まる。
④文字を挿入したいときは、その文字の一つ右側にカーソルを合わせ、 **INS** キーを押す。カーソルの形が変わり、この状態で文字を打ちこむと、何文字でも挿入することができる。挿入が終われば、もう一度 **INS** キーを押すか、カーソル移動キー、 **RETURN** キーのどちらかを押せば、カーソルが元に戻る。



▲挿入できる状態

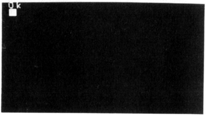
⑤ある行のすべてを変更したいときは、その行番号で新しいプログラムを入力し、 **RETURN** キーを押す。以前の行は、画面では残るが、パソコンの頭脳からは消える。
⑥ある行のすべてを削除したいときは、その行番号だけを入力して **RETURN** キーを押す。
⑦ある行(M)からある行(N)までを削除したいときは、「DELETE　M－N」と入力して **RETURN** キーを押す。
⑧ある行と行の間に新しい行を挿入するときは、前後の行番号の間の数字を行番号にして、新しいプログラムを打ちこむ。たとえば行100と行110の間には、行101～109の番号で挿入することができる。
⑨行番号を変更、つけなおししたいときは、**RENUM**（リナンバ）指令を使う。たとえば、最初の行番号15を100に変え、それ以降をすべて10刻みにするなら、「RENUM 100, 15, 10」と入力し、 **RETURN** キーを押す。これで、「GOTO～」など行番号に関係するプログラムも、すべて新しい行番号に対応させて変更される。

●――プログラムの実行と停止

プログラムをすべて入力し、まちがいがないことを確認したら、カーソルを何も文字が書かれていない行に移して、「RUN」 **RETURN** と押すと実行されます。この命令は、前述したようにファンクション・キーの **F5** にセットされています

から、F5 を一つ押すだけですみます。また、カーソルが、文字などのところにある場合は、F10 キーで実行することができます。

本書のゲームの多くは、終了するとプログラムの実行が終わり、「Ok」表示とカーソルが出て、次の命令（プログラムの入力、保存など）を受けつける状態になります。ただし、「お絵かきソフト」などは、そのままにしておくと、いつまでたってもプログラムの実行が終わりません。



▲Ok表示とカーソル

そういうときは、終了したいところで CTRL キーを押しながら STOP キーを押してください。「Break in ×××」と表示され（×××はストップしたところの行番号）、プログラムの実行が終わります。この方法は、プログラムの実行途中（ゲーム中など）で停止させたいときも同様です。ゲームを再開させるには、F8 キー、最初から再び実行するには F5/F10 キーを押します。

なお、プログラムの実行途中で STOP キーだけを押すと、ゲームなどが一時的にストップされ、もう一度押すと再開されます。

●──エラーが出たときの処理

プログラムの入力ミスや、命令の使い方などをまちがった場合、実行すると、エラーのあったところに対応して実行が中断し、メッセージとともにエラーのある行番号が表示されます。

そのときは、巻末につけたエラーメッセージの一覧表でエラーの意味を調べ、その行のプログラム・リストを F4 キーなどで画面に出し、エラー部分を調べて修正してください。

ただし、エラーの種類によっては、その行だけでなく前後の行も修正しなければならないことや、その行に関連するほかの行にまちがいがある場合もあります。

●──プログラムの保存と呼び出し

キーボードから打ちこんだプログラムは、本体内の記憶装置（メモリ）にしまわれます。これだけでは、電源を切るとすべて消えてしまいますから、カセットテープやフロッピィディスクなど補助記憶装置に保存しておきましょう。こうしておけば、いつでもすきなときに呼び出して、ゲームなどを楽しむことができます。

プログラムを補助記憶装置に保存することをセーブ、逆に、補助記憶装置に保存されたプログラムを呼び出して、本体の記憶装置に読みこむことをロードといいます。

◉カセットテープの場合

〈セーブ〉

①本体にテープレコーダを接続する（マニュアル参照)。

②新しいカセットテープを入れ、テープレコーダを録音状態にする。

③プログラムに名前をつける(6文字以内ならなんでもよい)。ここでは､「GAME 1」としておく。

④ CSAVE（シーセーブ） 命令を使い、次のようにキーインする。

CSAVE　"GAME 1"　RETURN

⑤テープがまわりはじめる。このとき、テープレコーダのカウンタの数値をメモしておく。画面が次のようになれば完了。

```
CSAVE "GAME1"
Ok
■
```

〈ロード〉

①本体にテープレコーダを接続する。

②プログラムの入ったテープをセットし、テープレコーダを再生状態にする。このとき、呼び出したいプログラムのカウンタの数値の少し前に、テープを合わせておく。

③ CLOAD（シーロード） 命令を使い、次のようにキーインする。

CLOAD　"GAME1"　RETURN

④テープがまわりはじめ、画面が次のようになれば完了。なお、読みこみたいプログラムの前にほかのプログラムが入っていて、コンピュータがそれを読みとばしたときは、「SKIP : GAME0」などという表示が出る。

```
CLOAD "GAME1"
Found : GAME1
Ok
■
```

◉フロッピィディスクの場合

〈セーブ〉

①初期化した（マニュアル参照）フロッピィディスクをセットする。

②プログラムに名前をつける（たとえば「GAME1」)。

③ SAVE（セーブ） 命令を使い、次のようにキーインする。

SAVE "GAME1" RETURN

④画面が次のようになれば完了。

```
SAVE "GAME1"
Ok
■
```

〈ロード〉

①プログラムの入ったフロッピィディスクをセットする。

② LOAD（ロード） 命令を使い、次のようにキーインする。

LOAD "GAME1" RETURN

③画面が次のようになれば完了。

```
LOAD "GAME1"
Ok
■
```

MSXゲーム&プログラミング

◉プログラムどおり入力し、１行ぶんが終わったら、必ず RETURN キーを押してください。

◉行99の注釈文は、ゲームの仮タイトルです。入力しておくと、プリンタで打ち出したリストを整理するときに便利です。

◉プログラム中、緑色の文字や線は、グラフィック・シンボルです。



神経衰弱

大勢で遊ぶトランプの代表格は、なんといっても「神経衰弱」です。だれでも知っているように、裏向けたカードをバラバラに並べ、順に数字が合うように2枚ずつめくっていくゲーム。「そうだ、たしかこのへんにあったはずだが……ウーム、どちらだろう」などと悩んだ経験は、だれもが持っているはずです。

神経をフルに使って、開いたカードの位置と数字を覚えていく。こんがらかってどうしようもないところから、こんな名前がつけられたのでしょう。キミもこのゲームにチャレンジし、逆にコンピュータの神経を衰弱させてみてください。

①開始

F5/F10

↓

②ゲーム

01 ～ 30

↓

③終了

カードがすべてそろえばゲーム終了

遊び方

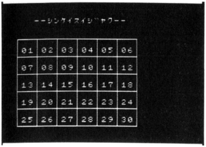
実行すると、縦5列×横6列の表があらわれる。このマス目の一つひとつが、いわばトランプのカードというわけ。カードには、01～30の番号がつけられているから、まずめくってみたいカードを選び、その数字をキーインする。



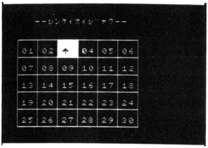
次に、同様にして、もう1枚のカードを選ぶ。2枚のカードが合っていなければ、2枚とも元に戻り、次の人の挑戦ということになる。

もし、2枚のカードがぴったり同じなら、カードがめくられたまま残り、次のカードを選ぶ権利が与えられる。

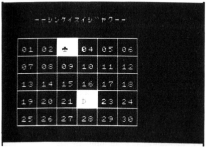
これをくり返し、30枚のカードのすべてがめくられればゲーム終了となる。



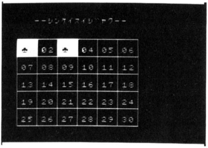
①最初の画面



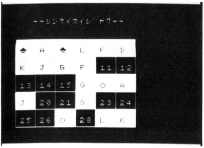
②1枚選ぶ



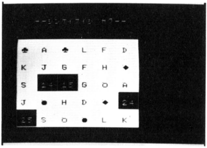
③もう1枚選ぶ。ハズレ



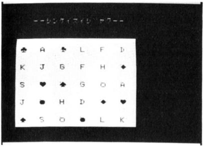
④アタリ



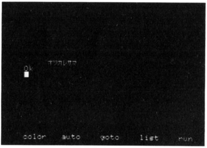
⑤かなり進んだぞ



⑥残り、あと4枚



⑦完成



⑧終了の画面

ゲームのコツと注意

◎カードをめくるための数字は、画面にあるように、必ず2桁(けた)でキーインすること（1、2ではなく01、02と押す)。もし、01を選ぶつもりで1を押してしまったようなときは、RETURN キーを押すとその1が消えるから、もう一度入れなおせばよい。

◎キーは、ポンと押してすぐに離すこと。長く押していると、次にとんでもない数字が選ばれてしまうこともある。

◎二人以上で遊ぶときは、だれがどのカードを合わせたかを、別にメモしておこう。そうしないと、勝ち負けがわからなくなるぞ。



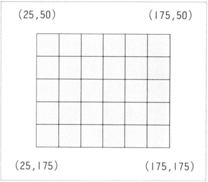
ゲームの改造

◘カードの模様を変える

同じカードばかり出てきてあきてしまったら、カードの絵文字を変えてみよう。行290の◆以下を、キミのすきなグラフィック・シンボルや、文字、数字に変えればよい。

ただし、選ぶのは15種類で、同じものを2回並べて合計30個にすること。

プログラムの解説



◆画面の初期設定〈行100～180〉

行100　使用画面を高解像度グラフィック画面に設定し、使用する色を決める。

行110　各配列変数を設定し、乱数の発生が実行のたびに変わるようにする。

行120　グラフィック画面に文字が表示できるようにする。

行130～150　カードわくの縦線を7本表示するループ。

行160～180　カードわくの横線を6本表示するループ。

◆タイトルとマス目番号の表示〈行190～270〉

行190　画面の上方に「シンケイスイジャク」とタイトルを表示する。

行200　マス目番号の表示用変数A$に、01～30の数字を入れる。

行210～230　マス目番号を表示するための座標を決める。各マス目の左上端のX座標に7、Y座標に10をプラスし、それぞれの値をS2とS1に入れる。

行240～260　01～30の数字を、順に各マス目に表示する。

行270　マス目番号のデータ。

◆カードの絵文字の作成〈行280～340〉

行280　READ（リード）文で、絵文字をD$に読みこむ。

行290　絵文字のデータ。15種類を2個ずつ用意する。

行300～340　乱数を発生させ、それを絵文字のデータと対応させることによって、カードを各マス目に振り分ける。具体的には、乱数を行310でZに入れ、配列変数D$のZ番めの絵文字をW$に入れる。そのとき、同じ絵文字が2回使われないよう、Wに1をセットしておく。行320でW(Z)が0かどうかを判断しているのは、そのためである。

◆カードの選択〈行350～480〉

行350　カードを2枚選ぶ(めくる)ためのループの開始。

行360　変数B$を、何もない状態にしておく。

行370～400　数字がキーインされるまで待ちつづけ(行370、390)、キーインされた内容をB$に入れる。

行410　入力された内容（B$）を数値に変換して、その数値が 0 より大きく、31より小さいかどうかをチェックする。もしその範囲外であれば、行360に戻る。

行420　すでに割り振られたカードかどうかをチェックする。

行430　1 回めにキーインされた数字を IH に入れる。

行440　2 回めにキーインされた数字を IW に入れる。

行450　2 回めの数字が 1 回めの数字と同じかどうかをチェックし、等しいときは、再度キーインを待つ（行360に戻る）。

行460　キーインされた数字のマス目に、割り振られたカード（絵文字）を表示する。

行470　選んだカードを、K$ に入れる。

行480　ループの終わり。

➡カードの判定 〈行490～550〉

行490　選んだ 2 枚のカードが等しいかどうかをチェックし、等しければ行500へ、そうでなければ行520へ飛ぶ。

行500　正答数 TI に 1 を加え、選んだ 2 枚のカードに 1（すでに割り振られたことを示す）をセットする。

行510　正答数が15未満のときは、カードの選択（行350）に戻る。TI が15になれば終了なので、行560へ飛ぶ。

行520～540　選ばれたカードをしばらく表示したあと、元に戻す。

行550　カードの選択（行350）に戻る。

➡終了の処理 〈行560～600〉

行560～590　画面に「オツカレサマ」と表示する。

行600　プログラムの実行を終わる。

使用変数

A$　01～30までの数字をしまう

D$　カードの絵文字をしまう

G　正解のカードに、すでに割り振られたことを示す 1 がセットされる

K　選んだカードが 1 枚めか 2 枚めかという番号が入る

S1　各マス目の数字を表示するためのＹ座標が入る

S2　各マス目の数字を表示するためのＸ座標が入る

TI　正解の組数

W　乱数発生させた数字を入れる

W$　キーインされた内容を入れる

```
99   'ｼﾝｹｲｽｲｼﾞｬｸ
100 SCREEN 2:COLOR 15,4,7:CLS
110 DIM A$(30),D$(30),W$(30),W(30),G(30),S1(30),S2(30),K(2):E=RND(-TIME)
120 OPEN "GRP:" AS #1
130 FOR I=0 TO 6
140  LINE (I*24+24,48)-(I*24+24,168),15
150 NEXT I
160 FOR I=0 TO 5
170  LINE (24,I*24+48)-(168,I*24+48),15
180 NEXT I
190 PSET(45,20),4:COLOR 15,4:PRINT#1,"--ｼﾝｹｲｽｲｼﾞｬｸ--"
200 FOR R=1 TO 30 :READ A$(R):NEXT R
210  FOR P1=1 TO 5 :FOR P2=1 TO 6
220    W=W+1:S2(W)=(6+(24*P2)):S1(W)=(36+(24*P1))
230 NEXT P2:NEXT P1
240 FOR P1=1 TO 5 :FOR P2=1 TO 6
250  X=X+1 :PSET (S2(X),S1(X)),4:COLOR15,4:PRINT#1, A$(X)
260 NEXT P2:NEXT P1
270 DATA 01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24
,25,26,27,28,29,30
280 FOR R=1 TO 30:READ D$(R):NEXT R
290 DATA ♦,♠,♥,♣,●,○,A,S,D,F,G,H,J,K,L,♠,♦,♣,♥,●,○,A,S,D,F,G,H,J,K,L
300 FOR J=1 TO 30
310  Z=INT(RND(1)*30+1)
320  IF W(Z)=0 THEN W$(Z)=D$(J) ELSE 310
330  W(Z)=1
340 NEXT J
350 FOR K=1 TO 2
360 B$=""
370 IN$=INKEY$:IF IN$="" THEN GOTO 370
380 B$=B$+IN$
390 IN$=INKEY$:IF IN$="" THEN GOTO 390
```

```
400 B$=B$+IN$
410 IN=VAL(B$): IF IN>0 AND IN<31 THEN 420 ELSE GOTO 360
420 IF G(IN)=1 THEN 360
430 IF K=1 THEN IH=IN
440 IF K=2 THEN IW=IN
450 IF K=2 AND IH=IW THEN 360
460 LINE  (S2(IN)-6,S1(IN)-12)-(S2(IN)+18,S1(IN)+12),15,BF:PSET (S2(IN),S1(IN))
:COLOR 4:PRINT#1, W$(IN)
470   K$(K)=W$(IN)
480 NEXT K
490 IF K$(1)=K$(2) THEN 500 ELSE 520
500 TI=TI+1:G(IH)=1:G(IW)=1
510 IF TI<15 THEN  350 ELSE 560
520 FOR T=0 TO 500 :NEXT T
530 LINE(S2(IH)-5,S1(IH)-11)-(S2(IH)+17,S1(IH)+11),4,BF:LINE(S2(IH)-6,S1(IH)-12)
-(S2(IH)+18,S1(IH)+12),15,B:PSET (S2(IH),S1(IH)),4:COLOR 15:PRINT#1, A$(IH)
540 LINE(S2(IW)-5,S1(IW)-11)-(S2(IW)+17,S1(IW)+11),4,BF:LINE(S2(IW)-6,S1(IW)-12)
-(S2(IW)+18,S1(IW)+12),15,B:PSET (S2(IW),S1(IW)),4:COLOR15:PRINT#1, A$(IW)
550 GOTO 350
560 FOR W=0 TO 2000:NEXT W
570 SCREEN 0:COLOR 15,4,8:CLS
580 LOCATE 5,10
590 PRINT "ｵﾂｶﾚｻﾏ"
600 END
```

スキーゲーム

急増するスポーツ人口。なかでもウインタースポーツの花形であるスキーは、テニスと並んで人気を二分しているスポーツだといえます。白銀の世界を舞うように滑走する――そのスリルと爽快感が、多くの人をスキーにひきつけるゆえんでしょう。

このスキーゲームは、画面のコースを旗を倒しながら滑っていくもの。指先の操作だけで、存分にテクニックを発揮することができます。フィンガーテクニックを駆使して、キミもトップレーサーに挑戦してください。

①開始

F5/F10

↓

②ゲーム

← →

↓

③終了

木にぶつかるか、コースをはずれればゲームオーバー

遊び方

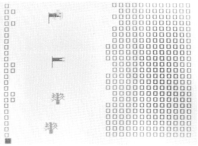
●ゲームの進め方

実行すると、画面の中央にコースができ、画面の下から旗や木がせまってくる。カーソル移動キーで人を左右に動かして、旗をとれば、旗門通過ということで100点プラスされる。

逆に、木にぶつかったり、コースからはずれると、ゲームオーバー。それまでに何点とったかを競う。



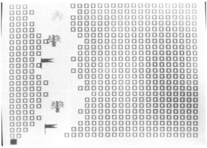
①最初の画面

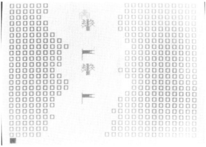


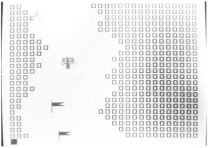
②旗をとる

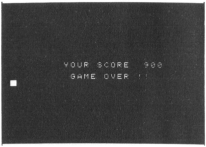
◉ゲームの終了

木にぶつかるか、コースを示す□□に少しでも触れると、その時点でゲームオーバー。スコアが表示され、自動的に終了となる。


③道が変化


④木にぶつかる


⑤コースをはずれる



⑥スコアの表示

ゲームのコツと注意



◎カーソル移動キーを小きざみに動かすこと。真横を通る木にはぶつからないから、木をやりすごしたらすぐに旗をめざして人を動かそう。

◎コースの幅が次々に変わるから、そちらにも注意しよう。木と旗ばかりに気をとられていると、コースからはずれてゲームオーバーになってしまうぞ。

プログラムの解説

▶初期設定〈行100〜140〉

行100　使用する画面のモード、色、桁数を決め、ファンクション・キーの表示を消す。

行110　変数をすべて整数型に宣言し、乱数の発生が実行のたびに変わるようにする。

行120　滑走コースをつくる関数を定義する。カタカナのロを使い、Xの数だけロを与える。

行130　乱数を発生させる関数を定義する。

行140　VRAMの内容を読みこみ、関数の値とする。

▶スプライト・パターンの定義〈行150〜240〉

行160〜220　人、旗、木のスプライト・パターンの定義。Jが0のときは人、1のときは旗、2のときは木。

行230　各変数の初期値を決める。得点に0、人の数に1、コースの幅に15、人のスタート位置に10をセット。

行240　スプライト・パターンを消すための初期設定。

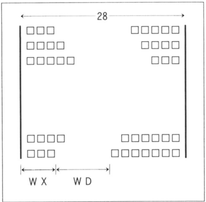
▶ゲームの開始〈行250〜390〉

行250　人と旗、人と木が衝突したときの、飛び先の指定（行470へ）。

行260　コースの幅、木や旗を出すかどうかを決める変数の値を設定する。

行270〜310　コースの幅を変化させるための処理。**TIME**変数を使い、Tの値を0〜10の範囲で変化させることによって、コースの幅を変える。

行320〜330　コースの表示。画面の最下位行に表示していくことによって、スクロールし、コースが動いているように見える。



行350〜380　カーソル移動キーの押された方向（→なら3、←なら7）を読みとり、人を移動させる。

行390　人がコースをはずれたかどうかをチェックする。人がいる場所の画面（VRAM）が空白(64)かどうかをチェックし、

空白でないときはコースをはずれたと判断する。

◯旗、木の表示と移動 〈行400〜460〉

行410〜450　コースの幅をあらわす変数 FNR(WD) と旗、木を出す変数 CC の値によって、旗と木を表示し、移動させる。

◯旗、木に人が衝突したときの処理 〈行470〜510〉

行470〜510　スプライトの割りこみを禁止し、人がどのスプライト・パターンと衝突したかをチェックする。木のときは、終了処理(行520) へ飛ぶ。旗のときは得点を100プラスし、旗を消す。

◯終了処理 〈行520〜550〉

行520　音楽を鳴らす。

行530〜540　スコアと「GAME OVER」を表示する。

行550　終了画面をしばらく表示して、プログラムの実行を終わる。

◯スプライト・パターンのデータ 〈行560〜610〉



使用変数

変数	意味	変数	意味
AD	コースの幅を変化させるための変数	FNR	乱数を発生させる関数
CC	旗、木を出す	LE	人の数
CX	人の座標	PO	得点
FNL$	コースをつくる関数	WD	コースの幅
FNP	コースをはずれたかどうかを判断する関数	WX	コース外の幅
		Y(I,J)	旗と木の位置

```
99 '  SKI ｹﾞｰﾑ
100 SCREEN 1,2,0:COLOR 2,15,15:WIDTH 29:KEY OFF
110 DEFINT A-Z:R=RND(-TIME)
120 DEFFNL$(X)=STRING$(X,"ロ")
130 DEFFNR(A)=RND(1)*A
140 DEFFNP(X,Y)=VPEEK(6146+X+Y*32)
150 '  ｽﾌﾟﾗｲﾄ･ﾊﾟﾀｰﾝ ﾉ ﾃｲｷﾞ ﾄ ｹﾞｰﾑ
160 FOR J= 0 TO 2
170   S$=""
180   FOR I= 1 TO 32
190     READ A$:S$=S$+CHR$(VAL("&h"+A$))
200   NEXT I
210   SPRITE$(J)=S$
220 NEXT J
230 PO=0:LE=1:WD=15:CX=10:LOCATE 0,10:TIME=0:CC=0
240 FOR I=0 TO 1: FOR J=0 TO 1 :Y(I,J)=209:NEXT J,I
250 ON SPRITE GOSUB 470 :SPRITE ON
260 AD=FNR(3)-1:CC=CC-1
270 WX=WX+AD
280 IF WX<1 THEN WX=1 :GOTO 300
290 IF WX+WD>26 THEN WX=WX-1:WD=26-WX
300 T=SGN(((TIME¥100) MOD 20)-10)
310 IF FNR(8)=0 AND ABS(WD+T-11)<7 THEN WD=WD+T
320 R$=FNL$(WX)+SPACE$(WD)+FNL$(28-WX-WD)
330 LOCATE 0,23:PRINT R$
340 '
350 S=STICK(0)
360 IF S=3 AND CX<25 THEN CX=CX+1
370 IF S=7 AND CX>1 THEN CX=CX-1
380 PUT SPRITE 0,(CX*8+16,8),7,2
390 IF FNP(CX,3)+FNP(CX+1,3)=64 THEN 410 ELSE 520
400 '  ﾊﾀ ﾄ ｷ ﾉ ﾋｮｳｼﾞ ﾄ ｲﾄﾞｳ
```

```
410 FOR J=0 TO 1:FOR I= 0 TO 1
420   IF Y(J,I)=209 AND FNR(WD)<5 AND CC<0 THEN CC=2:X(J,I)=(WX+(WD/2))*8:Y(J,I)
=192
430   IF Y(J,I)<8 THEN Y(J,I)=209 ELSE IF Y(J,I)<209 THEN Y(J,I)=Y(J,I)-10
440   PUT SPRITE 1+J*10+I,(X(J,I),Y(J,I)),3+5*J,J
450 NEXT I,J
460 GOTO 260
470 SPRITE OFF:FOR J1=0 TO 1:FOR I1=0 TO 1
480 IF Y(J1,I1)=209 OR ABS(X(J1,I1)-(CX*8+16))>18 OR ABS(Y(J1,I1)-8)>18 THEN 5
00
490 IF J1=0 THEN 520 ELSE PLAY "L64O6A":PO=PO+100:Y(J1,I1)=209:PUT SPRITE 1+J1
*10+I1,(0,209),1,1:SPRITE ON:RETURN
500 NEXT I1,J1
510 SPRITE ON:RETURN
520 PLAY "O4L32BAGFEDC"
530 SCREEN 1:COLOR 8,4,7:LOCATE 8,10:PRINT "YOUR SCORE ";PO
540 LOCATE 9,12:PRINT "GAME OVER !!"
550 FOR I=1 TO 3000:NEXT I:CLS:END
560 DATA 23,33,19,75,13,79,67,11,8F,55,23,43,03,03,03,03
570 DATA 90,94,BE,C4,8A,9B,E4,8C,92,A1,C0,C0,C0,C0,C0,C0
580 DATA 40,40,7F,7F,7F,7F,7F,40,40,40,40,40,40,40,40,40
590 DATA 00,00,FF,F8,C0,F8,FF,00,00,00,00,00,00,00,00,00
600 DATA 00,03,03,01,0f,37,23,2b,2b,2a,2a,2c,28,28,0c,00
610 DATA 00,c0,c0,80,f0,ec,c4,d4,d4,54,54,34,14,10,30,00
```

お絵かきソフト MSX2対応

幼いころ、12色のクレヨンを、はじめて手にしたときのうれしさを覚えていますか。絵を描くうれしさよりも、クレヨンを買ってもらったという喜びのほうが大きくて、よく友だちに見せびらかしたものでした。でも、そんなとき、友だちが自分よりも豪華な24色のクレヨンを持っていたりすると、うれしさがたちまち、にくらしさに変わってしまったものでした。

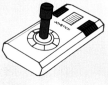
ほかのゲームにも少し飽きたな、なんて思ったら、幼い日を思い出しながら絵を描いてみるのもいかがでしょうか。画用紙代わりの画面に向かい、ジョイスティックというクレヨンを使って……。

①開始

F5/F10

⬇

②ゲーム
ジョイスティックとトリガボタン



⬇

③終了

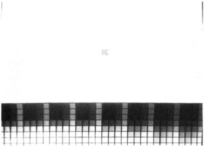
CTRL + STOP

遊 び 方

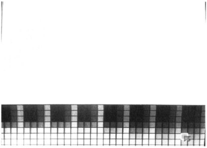
●画用紙とペン、絵の具

プログラムを実行すると、画面の下３分の１の部分に、色が表示されていく。色の数は、32×８＝256色。これがいわばパレットで、キミが使える色になる。

残りの３分の２の白い部分が画用紙で、真ん中に青い矢印が表示される。この



①最初の画面



②使いたい色を選ぶ

ペンを使って、画用紙に前後、左右、斜めにと、自由に絵を描いていくわけだ。

◉絵の描き方

まず、ジョイスティックでペンを使いたい色のところに持っていき、トリガボタン（1、2の2種類あるときは1。以下同じ）を押す。これで、ペンに色がついたことになる。このとき、選んだ色が白わくで囲まれ、自分の使っている色がひと目でわかるようになっている。

次に、ペンを画用紙の描きたいところに移動させ、トリガボタンを押す。これで、ペンが画用紙におりたことになる。そうして線を描いたら、色をかえたいところで、トリガボタンを押す。これで、ペンが画用紙から離れたことになるから、次の色を選ぶ。

このように、トリガボタンでペンを上げ下げして、絵を描いていくわけ。この手順をまとめると、次のようになる。

①スタート。ペンを使いたい色のところに移動。

②トリガボタンを押す。

③ペンを描きたいところに移動。

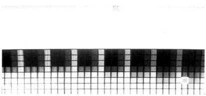
④トリガボタンを押す。

⑤ペンを移動させて、絵を描く。

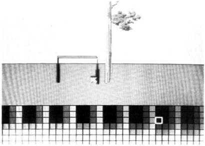
⑥トリガボタンを押す。

⑦次に使う色のところにペンを移動。

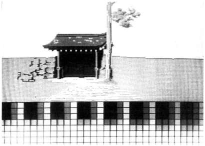
なお、同じ色を別の場所で使いたいときは、トリガボタンでペンを上げ、画用紙上を移動させてトリガボタンを押すと、そこから同じ色で描くことができる。



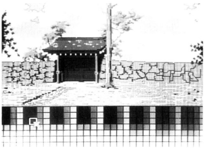
③位置を決めて線を描く



④絵を描いていく



⑤作画の途中

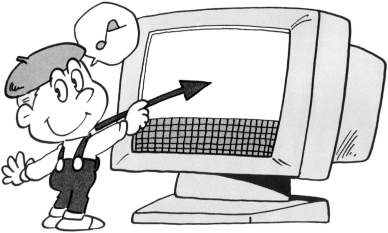


⑥完成

ゲームのコツと注意

◎描くのは線だけだから、ぬりつぶしたいときは、ペンを小きざみに上下左右に動かして、線や点でつぶしていくことになる。

◎はみ出した場合は消せないから、慎重にペンを動かすこと。注意してもはみ出してしまったときは、ほかの色を上から重ねれば前の色が消えるから、それを利用するとよい。



プログラムの解説

画面の初期設定と初期処理〈行100〜160〉

行100　変数の型を整数型に宣言し、画面のファンクション・キーの表示を消す。

行110　使用する画面を、ビットマップ・グラフィック・モードに設定する。

行120　ペンの移動範囲の座標をチェックする関数（X、Y）を定義する。

行130〜140　ジョイスティックが押された方向により、カーソルを移動させるための値（－1、0、1）を決める関数を定義する。

行160　各変数の初期値をセットする。

256色の表示〈行170〜260〉

行180〜260　画面の下段に、ゲームで使える256色を表示する。

ペンの作成と上げ下げ〈行270〜340〉

行280〜320　ペンの形をスプライト・パターンで作成する。

行340　ペンを上げ下げさせるため、トリガボタンが押されたとき、行490の処理へ飛ぶ。

▶ペンの表示〈行350〜420〉

行360　ペンのスプライト・パターンを表示する。

行370　ペンを移動する幅と下限の初期値を設定する。

行380　ジョイスティックのトリガボタンによる割りこみを許可する。

行390　ペンが上がった状態であれば、行410へ飛ぶ。

行400　ペンが256色表示の位置にあれば、色の選択の処理（行530）へ飛ぶ。そうでないときは、色を描く。

行410　ジョイスティックが押された方向を読みとる。

行420　ペンの、現在の座標を保存する。

▶ペンを移動するルーチン〈行430〜470〉

行440　ジョイスティックが押されていなければ、行360に戻り、ペンを表示したままにする。

行450　ジョイスティックが押された方向により、ペンを移動する座標を決める。

行460　ペンの移動先が作画画面からはずれたとき、ブザーを鳴らし、ペンの移動を無視する。

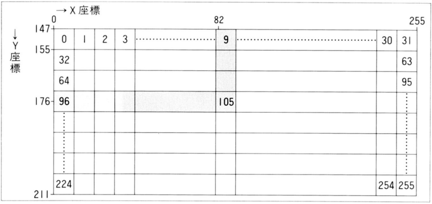
行470　行360に戻り、ペンを表示したままにする。

▶ペンを上下させるルーチン〈行480〜510〉

行490　ジョイスティックのトリガボタンによる割りこみを禁止する。

行500　ペンが上がっている場合は下げ、下がっている場合は上げる。

▶色の選択〈行520〜600〉



行530〜550　画面の下段に表示される256色は、前ページ下図のようにカラー番号が割りあてられ、1色は8×8の座標で構成されている。そして、X1にはトリガボタンが押された位置のX座標、Y1に同じくY座標がセットされている。

行530で、V1にそのときのX座標が左から何番めの色のマス目にあるか、行540で、V2にそのときのY座標のある位置の左端のカラー番号は何か、がそれぞれ代入され、行550によってトリガボタンが押された場所のカラー番号(V1+V2)がC1にセットされる。

たとえば、X座標が82、Y座標が176であれば、図のようにV1は9、V2は96で、C1に105がセットされる。

行560　前に選ばれた色の白わくを、黒に戻す。

行570　選ばれた色のマス目の左端のX座標とY座標を計算する。

行580　選ばれた色のマス目を、白わくで囲む。

行590　ペンを上げる。

▶ペンのスプライト・パターンのデータ〈行610〜620〉

行620　スプライト・パターンのデータ文(16進表記)。

使用変数

BS	色のマス目の1枚の幅	PA$	ペンの状態をセットする
BX	256色表示欄の左端のX座標	S	ジョイスティックが押された方向
BY	256色表示欄の左端のY座標	XP	256色各マス目の左端のX座標
C	ペンの移動の下限値	YP	256色各マス目の左端のY座標
C1	選択された色のカラー番号	Z	ペンの移動幅

●ジョイスティックがないときは

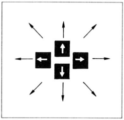
次のようにプログラムを変更することによって、ジョイスティックの代わりにスペース・キーとカーソル移動キーが使えるようになる。

①行340の「GOSUB」に続くコンマ(，)をとる。

②行380、410、490の「STRIG」または「STICK」に続くカッコ内の1を、0にかえる。

➜トリガボタン→スペース・キー［　　　］

➜方向→カーソル移動キー（斜めは、その方向の両側のキーを同時に押す。右図参照）



```
99 'おえかきゲーム
100 DEFINT A-Z:KEY OFF
110 SCREEN 8,2
120 DEFFNP(X,Y,C)=NOT((X>=0) AND (X<256) AND (Y>=0) AND (Y<C))
130 DEFFNX(S)=VAL(MID$("112221000",S+1,1))-1
140 DEFFNY(S)=VAL(MID$("100122210",S+1,1))-1
150 '     ショキショリ
160 BX=0:BY=147:BS=8:X1=128:Y1=74:PA$="UP"
170 '     256ショク ノ ヒョウジ
180 LINE(0,0)-(255,211),255,BF
190 FOR I=0 TO 255
200     XP=I MOD 32
210     XP=BX+XP*BS
220     YP=I ¥   32
230     YP=BY+YP*BS
240     LINE(XP,YP)-STEP(BS,BS),1,B
250     LINE(XP+1,YP+1)-STEP(BS-1,BS-1),I,BF
260 NEXT I
270 '     ペン サクセイ
280 FOR I=1 TO 8
290      READ P$
300      S$=S$+CHR$(VAL("&H"+P$))
310 NEXT I
320 SPRITE$(0)=S$
330 '     ペン ノ アゲサゲ
340 ON STRIG GOSUB,490
350 '     ペン  ヒョウジ
360 PUT SPRITE 0,(X1,Y1),9,0
370 C=211:Z=5
380 STRIG(1) ON
390 IF PA$="UP" THEN 410
```

```
400 IF Y1>146  THEN GOSUB 530 ELSE C=146:Z=1:PSET(X1,Y1),C1
410 S=STICK(1)
420 X=X1:Y=Y1
430 '     ﾍﾟﾝ  ｲﾄﾞｳ
440 IF S=0 THEN GOTO 360
450 X=X+(FNX(S)*Z):Y=Y+(FNY(S)*Z)
460 IF FNP(X,Y,C) THEN BEEP ELSE X1=X:Y1=Y
470 GOTO 360
480 '     ﾍﾟﾝ ﾉ ｼﾞｮｳｹﾞ
490 STRIG(1) OFF
500 IF PA$="UP" THEN PA$="DW" ELSE PA$="UP"
510 RETURN
520 '     ｲﾛ ﾉ ｾﾝﾀｸ
530 V1=INT(X1/8):PLAY "C"
540 V2=INT((Y1-147)/8)*32
550 C1=V1+V2
560 IF SY<>0 THEN LINE(SX,SY)-STEP(BS,BS),0,B
570 SX=(C1 MOD 32)*BS+BX:SY=(C1¥32)*BS+BY
580 LINE(SX,SY)-STEP(BS,BS),255,B
590 PA$="UP"
600 RETURN
610 '     ﾍﾟﾝ ﾉ ﾃﾞｰﾀ
620 DATA FF,FE,FC,FC,FE,FF,CF,86
```

花火大会 MSX2対応

夏の風物詩である花火。涼風感のある打ちあげ音とともに、人によってはスイカ、アイスクリーム、ビールなど、いろんなものを連想することでしょう。

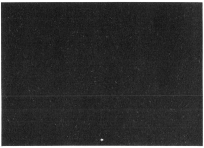
そんな花火を、一年中、しかも自宅にいながら楽しめるようにくふうしたのがこのゲーム。夏の夜空を思わせる画面に、赤、青、黄、緑などがクロスオーバーされ、花火が美しく開花します。しばらくはプログラムのことも忘れ、ゆっくり花火観賞にひたってください。

①開始
F5/F10
↓
②ゲーム
↓
③終了

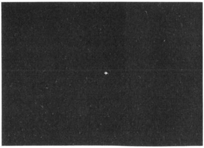
遊び方

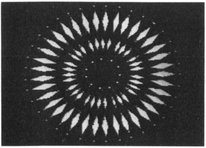
実行すると、夜空を思わせるように、画面が黒一色になる。何も出てこないからといって、あせる必要はない。いつ打ちあげられるかと待つことも、花火の楽しみのひとつ。この状態で1分間、ただひたすらに待つ。

しばらくすると、画面中央の下から花火の打ちあげが始まる。あとは、色とりどりに咲いては散る、花火の美しさを楽しむだけだ。

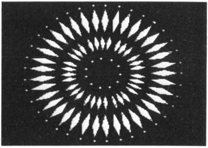

①最初の画面




②打ちあげ開始



③花火がうっすらあらわれる



④ドーンと開花

プログラムの解説

▶初期設定〈行100～130〉

行100 使用する画面を、ビットマップ・グラフィック・モードにする。

行110 色の指定。背景色、周辺色を黒にする。

行120 使用する変数の、初期値を設定する。

行130 パレット番号4、5、7、8をすべて黒に指定する。これにより、行140～170で花火を描く過程を見えなくする。

▶花火の形を描く〈行140～170〉

行140 花火を形づくるいちばん内側の円を描くため、変数を設定して円を描くサブルーチン（行570）を呼び出す。

行150～160 2番め、3番めの円をひし形で描くため、サブルーチン（行420）を呼び出す。

行170 4番めの円を描く。

▶花火を打ちあげる軌跡を描く〈行180～240〉

行180 Y座標を20ドットずつ減らしていくことによって、打ちあげの軌跡を描くループ。

行190～200 Y座標の値により、そこを中心に半径1の円を赤で描き、赤でぬりつぶす。

行210 描いた円を見るための時間かせぎ。

行220～230 行190～200で描いた円と同じものを描き、黒でぬりつぶす。

行240 Y座標が110になるまでくり返し、ループを終わる。

▶花火に色をつけて、音を出す〈行250～400〉

行250～270 色のR(赤)、G(緑)、B(青)の値を変えるループ。

行280〜310　パレット番号4、5、7、8にRGBの値を与える。花火に、透明（見えない）から色をつけることによって見えるようにする。

行330〜350　花火の爆発音をつくる。

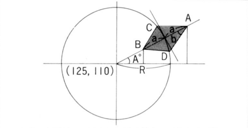
行360〜370　音の長さを決め、音を消す。

➡花火を形づくるひし形を描く〈行410〜550〉

行420　ひし形を描くループの開始。変数Tの値の数だけ、ひし形をつくる。

行430〜480　ひし形を描くための三角関数の計算。

行490〜520　**LINE**（ライン）文で、ひし形の4辺をつくる。しくみは、図のようにA〜D点を求め、それを線で結ぶ。ただし、C、Dが円周上の点であるため、正確なひし形ではないが、角度bが小さいため、画面上ではその差がわからない。



A点＝(125＋a×R×cosA°, 110＋a×R×sinA°)
B点＝(125−a×R×cosA°, 110−a×R×sinA°)
C点＝(125＋R×cos(A°＋b), 110＋R×sin(A°＋b))
D点＝(125＋R×cos(A°−b), 110＋R×sin(A°−b))

行530　ひし形に色をぬる。

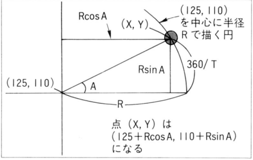
➡花火を形づくる円を描く〈行560〜630〉

行570　円を描くループの開始。変数Tの値の数だけ円をつくる。

行580〜590　円を描くための三角関数の計算。

行600　**CIRCLE**（サークル）文で円を描く。各円の中心座標（X，Y）は、図のように与えられる。

行610　円に色をぬる。



使用変数

- **A**　ひし形、円のバラツキの程度
- **C**　ひし形、円を描くときのパレット番号
- **H**　**CIRCLE**（サークル）文で円を描くときの半径
- **PB、PG、PR**　各パレットのＲＧＢの値
- **R**　ひし形、円を描くときの半径
- **T**　ひし形、円を描くときの個数
- **X**　X座標
- **Y**　Y座標

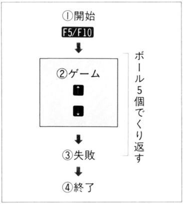
```
99 ' はなび゛の シュミレート
100 SCREEN 5
110 COLOR 15,1,1
120 X=125:Y=110:H=1
130 COLOR=(4,0,0,0):COLOR=(5,0,0,0):COLOR=(7,0,0,0):COLOR=(8,0,0,0)
140 A=16:R=20:T=11:C=4:GOSUB 570
150 A=30:R=40:T=31:C=5:GOSUB 420
160 A=18:R=70:T=37:C=7:GOSUB 420
170 A=18:R=90:T=38:C=8:GOSUB 570
180 FOR Y=210 TO 110 STEP -20
190   CIRCLE (125,Y),1,6
200   PAINT  (125,Y),6
210   FOR X=0 TO 100:NEXT X
220   CIRCLE (125,Y),1,1
230   PAINT  (125,Y),1
240 NEXT Y
250 FOR PR=0 TO 7
260   FOR PG=0 TO 7
270     FOR PB=0 TO 7
280       COLOR=(4,PR,PG,PB):FOR M=1 TO 70:NEXT M
290       COLOR=(5,PR,PG,PB):FOR M=1 TO 70:NEXT M
300       COLOR=(7,PR,PG,PB):FOR M=1 TO 70:NEXT M
310       COLOR=(8,PR,PG,PB):FOR M=1 TO 70:NEXT M
320     NEXT PB
330    SOUND 0,112:SOUND 1,13:SOUND 2,3:SOUND 3,10:SOUND 4,54
340    SOUND 5,10:SOUND 6,23:SOUND 7,198:SOUND 8,31:SOUND 9,31
350    SOUND 10,25:SOUND 12,75:SOUND 11,189:SOUND 13,8
360    FOR J=0 TO 700 :NEXT J
370    FOR J=1 TO 13:SOUND J,0:NEXT J
380   NEXT PG
390 NEXT PR
400 END
```

```
410 '
420 FOR S=1 TO T
430   C1=COS(S*A)
440   S1=SIN(S*A)
450   C2=COS(S*A+.05)
460   S2=SIN(S*A+.05)
470   C3=COS(S*A-.05)
480   S3=SIN(S*A-.05)
490   LINE (X+.8*R*C1,Y+.8*R*S1)-(X+R*C2,Y+R*S2),C
500   LINE (X+R*C2,Y+R*S2)-(X+1.2*R*C1,Y+1.2*R*S1),C
510   LINE (X+1.2*R*C1,Y+1.2*R*S1)-(X+R*C3,Y+R*S3),C
520   LINE (X+R*C3,Y+R*S3)-(X+.8*R*C1,Y+.8*R*S1),C
530   PAINT (X+R*C1,Y+R*S1),C,C
540 NEXT S
550 RETURN
560 '
570 FOR S=1 TO T
580   X1=COS(S*A)
590   Y1=SIN(S*A)
600   CIRCLE (X+R*X1,Y+R*Y1),H,C
610   PAINT  (X+R*X1,Y+R*Y1),C
620 NEXT S
630 RETURN
```

スカッシュを決めろ!!

いまや日本は、爆発的なスポーツブーム。スカッシュは、まだそれほど知られていないものの、これからの普及の可能性が、かなり見こまれているようです。

スカッシュは、イギリスで生まれた室内競技。テニスの壁打ち練習をゲームにしたようなもので、二人の選手が壁に向かって交互にボールを打ち合うという、意外にハードなスポーツです。



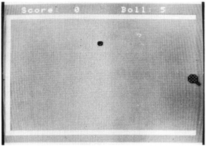
このパソコンで遊ぶスカッシュは、コンピュータ相手のいわば“一人スカッシュ”。交互に打ち合う感じはありませんが、得点が高くなるにつれて途中でボールが消えたり、障害物があらわれたりと、コンピュータならではのくふうが随所(ずいしょ)にこらされています。実際のゲームのように、全身から汗ということはないにしても、文字どおり手に汗を握る興奮は味わえるはずです。

遊び方

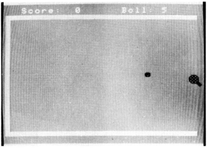
◉ゲームの進め方

実行すると、壁、コート、ラケットが表示され、すぐにボールが左から右に飛んでくる。キミは、右端の赤いラケットを上下に動かして、ボールをうまく打ち返そう。

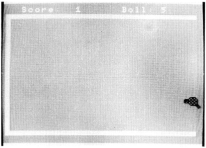
ゲームで使うキーは、カーソル移動キーのうち、↑↓の二つだけ。ボールを1回打ち返すごとにスコアが1点プラスされる。1回のゲームで使えるボールの数は5個。1回ミスするとボールが1個減り、全部なくなると、ゲームオーバーとなる。しかし、逆に考えると、5個のボールがなくなるまで、いくらでも高得点をあげることが可能。キミの腕しだいというわけだ。



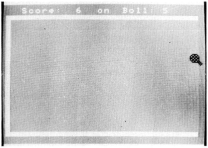
①最初の画面



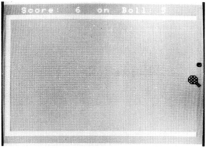
②ボールが飛んでくる



③ボールをラケットで打ち返す



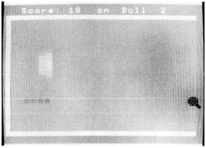
④6点とったら、画面中央でボールが消える



⑤残念、失敗



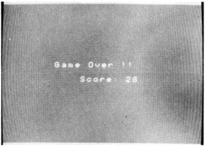
⑥11点で障害物出現



⑦16点以上で障害物が縦に動く



⑧21点。動く障害物が見えなくなる



⑨ゲームオーバー。スコアの表示

●スコアとボールの動き

とはいっても、得点が高くなるほどむずかしくなるのは、こういったゲームではお決まりのコース。次のように、スコアが6、11、16、21点以上になると、敵のいろんなしかけがあらわれる。

得　点	し　か　け
0～5点	ボールとラケットだけ
6～10点	ボールが画面の中央にくると、消えてしまう。これ以降、2点追加するごとに、ボールの消える範囲が広がっていく
11～15点	コートの左に障害物があらわれる
16～20点	障害物が縦に動く
21点以上	障害物が見えないまま縦に動く

ゲームのコツと注意

◎ラケットのどこか一部にでもボールが当たればよいから（グリップの根もとなどでもよい）、ネットの部分に当てようとあせることはない。ただし、ラケットはそんなに速く移動しないから、ボールのきそうな位置を考えて、ラケットを持っていくことが必要だ。

◎スコアが6点以上になり、ボールが画面中央で消えたときのコツは二つある。一つは、左の壁の近くにあって見えるときの動きで次にあらわれるコースを予測すること。もう一つは、ある音がヒントになる、とだけいっておくから、自分で考えてみよう。



プログラムの解説

▶初期設定〈行100～130〉

行100　使用画面を高解像度グラフィック・モードにし、スプライト・パターンの大きさを16×16に設定する。画面の色と桁（けた）数を決め、ファンクション・キーの表示を消す。

行110　変数をすべて整数型に宣言し、乱数の発生が実行のたびに変化するようにする。

行120　関数を定義する。

行130　グラフィック画面に文字を表示できるようにする。

⊃スプライト・パターンのセット 〈行140〜210〉

行150　スプライト・パターン作成のためのループの開始。

行160　スプライト・パターンを入れる変数S$を何もない状態にする。

行170　スプライト・パターンのデータ読みこみのためのループ。

行180　スプライト・パターンのデータ（行700〜720）を読みこみ、S＄に入れる。

行190　データ読みこみのためのループの終わり。

行200　S＄をスプライト・パターンにセットする。

行210　スプライト・パターン作成のループの終わり。

⊃ゲームの準備 〈行220〜250〉

行230〜240　ゲームに使う各変数の初期値をセットする。

行250　ゲーム開始の合図の音を出す。

⊃壁の表示 〈行260〜300〉

行270〜300　四角形を描く **LINE**（ライン）文で、三方の壁とコートをつくる。

⊃スプライト衝突時の割りこみ先の指定 〈行310〜320〉

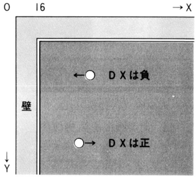
⊃スコアとボール数の表示 〈行330〜350〉

行340〜350　スコアと残りのボール数を表示する。

⊃スプライトの表示 〈行360〜400〉

行370　ラケットを明るい赤で表示する。

行380　ボールを表示する。ボールの色ＢＬが背景色と同じときは、ボールが消える。



行390　ラケットとボールを離すため、スプライトの衝突が起きてから（衝突が起こるとTM＝0となる）行370〜600を衝突による割りこみなしに実行する。

行400　スコアが10以上になれば、障害物を表示する。

⊃ボールを動かす 〈行410〜430〉

行420　ボールの座標を計算する。

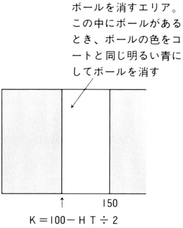
行430　ボールのX座標が16より小さくなったら、音を出し、X座標の増しぶんの正負を逆にする。前ページの図のように、ボールが壁に向かって進むとき、X座標の増しぶんDXは負の値をとり、ラケットに向かうときは正の値をとる。こうすることで、ボールは左の壁に当たったとき、はね返る。

▶ボールを消す〈行440〜510〉

行450　スコアの変数HTが6以上のとき、「on」と表示する。HTが5以下のときは、行500へ飛ぶ。

行460　ボールのX座標が50を超えたとき、ボールを消すエリアの左端のX座標Kを計算する。スコアが大きくなるにしたがって（2点ごと）、Kの値が小さくなり、ボールが消える範囲が大きくなっていく。

X座標が50以下のときは、ボールの色を緑にして、行500へ飛ぶ。



行470〜490　ボールのX座標がボールを消すエリアに入っているかどうかをチェックする。入っていればボールを明るい青にして、入っていなければ緑にする。

行500　ボールのX座標が250を超えたら音を出し、ボールの残り数を1マイナスし、次のボールを出すまで時間をおく。ボールの残り数が0になれば、行680の終了処理へ飛び、ボールが残っていれば行240へ飛ぶ。

行510　ボールのY座標が20より小さいか165より大きくなれば、音を出し、Y座標の増しぶんDYの正負を逆にする。こうすることで、ボールは上下の壁に当たったとき、はね返る（行430参照）。

▶障害物の制御〈行520〜540〉

行530　スコアが0以上で6より小さいとき、行560に飛ぶ。スコアが21以上になれば、障害物の色をコートと同じにして消す。そうでないときは、障害物の色を白にする。

行540　スコアが15より大きいとき、障害物を動かすために、そのY座標を計算しなおす。また、障害物のY座標が30より小さいか150より大きくなれば、座標の増しぶんDZの正負を逆にして、壁に当たったとき、はね返るように処理する（行430、行500参照）。

◆ラケットの制御〈行550〜600〉

行560　カーソル移動キーが、どの方向に押されているかを調べる。

行570　現在のラケットの位置を保存しておく。

行580　↑キーが押されたら、ラケットのY座標を10マイナスする。

行590　↓キーが押されたら、ラケットのY座標を10プラスする。

行600　ラケットのY座標が5より大きく180より小さいときは、そのまま行370へ飛ぶ。その範囲外のときは、Y座標の値を元に戻してから行370へ飛ぶ。

◆スプライトの割りこみ処理〈行610〜630〉

行620　スプライトが衝突しても割りこみが起こらないようにして、ボールがラケットに当たったとき、音を出し、スコアを1プラスする。

行630　ボールの進行方向を壁向きにして、角度を変え、しばらくはスプライトの衝突が起きないよう、TMを0にセットする。そうして、行420に戻る。

◆スコアの表示〈行640〜660〉

行650　スコアの表示欄をつくる。

行660　スコアを表示する。

◆終了処理〈行670〜680〉

行680　「Game Over」の文字とスコアをしばらく表示し、プログラムの実行を終わる。

◆スプライト・パターンの定義〈行690〜720〉

行700〜720　各スプライト・パターン（0がラケット、1がボール、2が障害物）のデータを16進であらわす。

使用変数

- **BL**　ボールの色
- **DX**　ボールのX座標の増しぶん
- **DY**　ボールのY座標の増しぶん
- **DZ**　障害物のY座標の増しぶん
- **HT**　ゲームのスコア
- **K**　ボールを消すエリアの指定
- **PL**　ボールの残り数
- **S**　スティック関数の値をセットする範囲
- **TM**　スプライトが衝突するタイミング
- **W**　時間かせぎの変数
- **XB**　ボールのX座標
- **XR**　ラケットのX座標
- **Y**　ラケットのY座標を保存するための変数
- **YA**　障害物のX座標
- **YB**　ボールのY座標
- **YR**　ラケットのY座標
- **YZ**　障害物のY座標

```
99 '          Squash   tennis   game
100 SCREEN 2,2:COLOR 15,4:WIDTH 29:KEY OFF
110 DEFINT A-Z:R=RND(-TIME)
120 DEF FNR(X)=RND(1)*X-1
130 OPEN"GRP:" FOR OUTPUT AS #1
140 '          スプライト パターン セット
150 FOR I=0 TO 2
160   S$=""
170   FOR J=1 TO 32
180     READ A$:S$=S$+CHR$(VAL("&h"+A$))
190   NEXT J
200   SPRITE$(I)=S$
210 NEXT I
220 '          ショキ ショリ
230 PL=5:HT=0:BL=2:ZC=5
240 XB=96:YB=32:XR=230:YR=96:DX=8:DY=6:YA=50:YZ=96:DZ=6
250 CLS:PLAY "o4l64cdef"
260 '          カベ ヒョウジ
270 LINE(10,15)-(240,20),11,BF
280 LINE(10,175)-(240,180),11,BF
290 LINE(10,15)-(15,175),11,BF
300 LINE(16,21)-(240,174),5,BF
310 '          スプライト ワリコミサキ
320 ON SPRITE GOSUB 620:SPRITE ON
330 '          ショキ スコア ボール ヒョウジ
340 PSET(30,5),4:PRINT #1,USING"Score:###";HT
350 PSET(150,5),4:PRINT #1,USING"Boll:##";PL
360 '          スプライト ヒョウジ
370 PUT SPRITE 0,(XR ,YR),9,0
380 PUT SPRITE 1,(XB,YB),BL,1
390 IF TM<3 THEN TM=TM+1 ELSE SPRITE ON
400 IF HT>10 THEN PUT SPRITE 2,(YA,YZ),ZC,2
```

```
410          ボール  ノ  セイギョ
420 XB=XB+DX:YB=YB+DY
430 IF XB<16 THEN PLAY "c":DX=-DX:XB=XB+DX
440 '        ボール  ヨケス
450 IF HT> 5  THEN PSET(120,5),4:PRINT #1,"on" ELSE GOTO 500
460 IF XB>50  THEN K=100-HT/2 ELSE BL=2:GOTO 500
470 IF XB =< K   THEN BL=2
480   IF XB > K   THEN BL=5
490     IF XB > 150 THEN BL=2
500 IF XB>250 THEN PLAY "bagfedcb":PL=PL-1:FOR W=0 TO 3000:NEXT:IF PL=0 THEN GOT
O 680 ELSE GOTO 240
510 IF YB<20 OR YB>165 THEN PLAY "G":DY=-DY:YB=YB+DY
520 '        ショウガイ  ブツ  セイギョ
530 IF HT=>0  AND HT<6  THEN GOTO 560 ELSE IF HT=>21 THEN ZC=5 ELSE ZC=15
540 IF HT >15 THEN YZ=YZ+DZ: IF YZ<30 OR YZ>150 THEN PLAY "ACEG":DZ=-DZ:YZ=YZ+DZ
550 '        ラケット  ノ  セイギョ
560 S=STICK(0)
570 Y=YR
580 IF S=1 THEN YR=YR-10
590 IF S=5 THEN YR=YR+10
600 IF(YR>5 )AND(YR<180) THEN GOTO 370 ELSE YR=Y:GOTO 370
610 '        スプライト  ワリコミ  ショリ
620 SPRITE OFF:IF NOT(XB>30 AND XB<70) THEN PLAY "BE":HT=HT+1:GOSUB 650
630 DX=-DX:DY=(DY+YB-YR-10)/2+FNR(3):TM=0:RETURN 420
640 '        スコア  ヒョウジ
650 LINE(80,5)-(105,13),4 ,BF
660 PSET(30,5),4:PRINT #1,USING"Score:###";HT:RETURN
670 '        シュウリョウ  ショリ
680 SCREEN 2,2:PSET (70,80),4:PRINT#1,"Game Over !!":PSET(100,100),4:PRINT#1,USI
NG"Score:###";HT:FOR G=0 TO 15000:NEXT:CLOSE #1:END
690 '        キャラクタ  パターン  データ
700 DATA 1f,26,66,99,99,e6,e6,99,99,66,26,1f,00,00,00,00,80,40,60,90,90,70,70,90
```

```
,90,70,58,f4,2a,15,0a,04
710 DATA 00,3c,7e,7a,7a,66,3c,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,00
720 DATA 00,00,00,00,ff,00,ff,00,ff,00,ff,00,00,00,00,00,ff,00,ff,00,ff,00,00,00
,00,00,ff,00,ff,00,ff,00
```

おたすけゲーム

①開始
F5/F10
↓
②ゲーム
←→
↓
③終了

ミスターX「レシーバー、レシーバー、応答せよ、応答せよ。さきほど、われわれの敵であるワルサーから、次のような挑戦状がまいこんだ」
――捕虜と盗んだ大金は、お望みどおり返してやろう。ただし、捕虜も大金もこの銀河系のあちこちの星に散らばっているので、いつ、どこからそちらに届くかわからない。レシーバー君、せいぜいカンを働かせておくことだな。おっと、いいわすれたが、キミたちがキャッチするのは、ありがたいものだけではない。よく覚えておくことだな。ハ、ハ、ハハハ……――
レシーバー「ミスターX、ありがたくないものって……？」
ミスターX「うーん、まさか……」
レシーバー「まさかって、あのまさかじゃないでしょうね」
ミスターX「なには、ともあれ、キミだけが頼りだ。成功を祈る！」
レシーバー「△○×◎！？」

遊び方

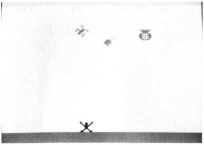
◉ゲームの進め方

実行すると、画面があらわれ、すぐに人、大金、爆弾が次から次へと落ちてくる。緑の帯が地面で、そこに立って両手を左右に広げているのがレシーバー、つまりキミだ。

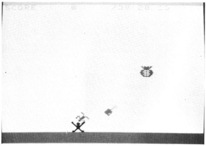
人をキャッチすると、100点。大金なら200点がプラスされるが、爆弾に当たった場合は、500点マイナスされて、しかもゲームオーバーになる。カーソル移動キーでレシーバーを左右に動かして、人と大金をうまくキャッチしよう。

◉人の残り数

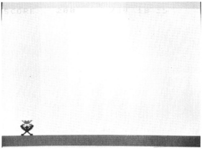
画面の右上方に「ノコリ　×人」の表示が出る。これは、20から受けそこなった人の数を引いた数で、ゲーム開始時には20人と表示される。これが0になれば、たとえ爆弾に当たらなくてもゲームオーバー。つまり20人以上受けそこなうことはできないわけだ。



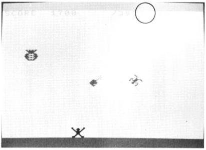
①最初の画面



②人をキャッチ

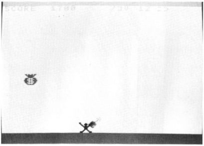


③大金をキャッチ



④人の残り数





⑤爆弾をキャッチ



⑥ゲームオーバー

ゲームのコツと注意

◎大金をキャッチすると人の2倍の得点になるが、人を20人受けそこなうと、ゲームオーバーになる。そこで、人を最優先にして、チャンスがあれば大金をねらう、という方法がよい。逆にいうと、人さえうまくキャッチしていれば、いくらでも得点をかせげるわけだ。

◎落下物のスピードは、人、爆弾、大金の順に速い。これを利用して、人を受けたらすぐに大金もキャッチ、ということが可能だ。

◎レシーバーの動きは、それほど速くない。レシーバーの位置から遠いところに人が落ちてきたときは、あきらめて大金をねらおう。

ゲームの改造

落下物のスピードを変える

行320の「DDH=」に続く数値によって、爆弾=8、大金=5、人=10と落下スピードが設定されている。この数値を大きくすれば、スピードが速くなり、小さくすれば遅くなる。慣れてくれば、どんどん速くするとよい。

プログラムの解説

初期設定〈行100〜120〉

行100　使用する画面を高解像度グラフィック・モードにし、色を決め、ファンクション・キーの表示を消す。

行110　すべての変数を整数型に宣言し、グラフィック画面に文字が表示できるようにする。

行120　各変数の初期値を設定し、配列変数を宣言する。

スプライト・パターンの作成〈行130〜190〉

行130　スプライト・パターン作成のためのループ開始。

行140　変数S$を何もない状態にする。

行150〜170　スプライト・パターンのデータを読みこむループ。

行180　スプライト・パターンのセット。スプライト番号0がレシーバー(受け手)、1が爆弾、2が大金、3が人になる。

行190　4回くり返して、ループを終わる。

画面の作成〈行200〜230〉

行200　各スプライト・パターンのY座標の基準位置を、初期値としてセットする。また、スプライト・パターンの表示スイッチを0（表示しない）にしておく。

行210　地面を暗い緑色でつくる。

行220　ゲーム開始時のスコアと人間の残り数を表示する。

行230　スプライトの衝突が起きたとき、行430へ飛ぶ。

▶レシーバーの移動 <行240～280>

行250　カーソル移動キーがどの方向に押されたかを調べ、■（A＝3）ならレシーバーのX座標を移動ぶんだけプラスし、■（A＝7）なら、移動ぶんだけマイナスする。

行260～270　レシーバーのX座標が255を超えるか、0より小さくなったら、レシーバーの動きを左右逆にする。

行280　レシーバーを表示する。

▶落下物の移動 <行290～410>

行300　落下物移動のためのループ開始。

行310　スプライト・パターンが表示されていないとき、落下開始地点のX座標を決める乱数を発生させ、スプライト・パターンを表示する。

行320　落下物のスピードを、爆弾＝8、大金＝5、人＝10にセットする。

行330　落下物が表示されていれば、そのスピードによってY座標を計算する。

行340　落下物のY座標が170を超えれば、行370へ飛ぶ。

行350　移動した落下物を表示する。

行360　落下物を、そのスピードに合わせて進める。

行370～380　落下物のY座標が170を超えたら、スプライト・パターンを消し、ループを終わる。

行390　スプライト・パターンが消された場合、表示のスイッチをオフ（0）にして、Y座標を17にする。

行400　落下する人をレシーバーが受けたとき、変数Kを0にする。人を受けそこなったとき、受けそこなった人数ERに1をプラスし、残りの数を表示する処理（行570）へ飛ぶ。ERがMH（20）より大きくなれば、ゲームオーバーの処理（行620）へ飛ぶ。

行410　行240に戻り、ゲームをくり返す。

▶落下物を受けたときの判断と処理 <行420～510>

行430　スプライトの衝突による割りこみを禁止する。

行440　それぞれのスプライト・パターンのＹ座標を調べ、144より小さい場合(衝突が起こっていない）は、行500へ飛ぶ。

行450〜470　落下物のスプライト・パターンとレシーバーのスプライト・パターンの座標を調べ、Ｘ座標が16以内の範囲にあり、しかも落下物のＹ座標が143以上であれば、スプライトが衝突したとして、行480へ飛ぶ。

行480　衝突した（受けとった）ものが爆弾なら、音楽を鳴らし、スコアを500マイナスして行610の終了処理へ飛ぶ。大金なら、音を出してスコアを200プラスし、大金の表示を消す（レシーバーだけ残す)。人なら、音を出してスコアを100プラスし、人の表示を消す。

行490　スコアの表示処理（行520）をし、衝突したスプライト・パターンを消す。

行500　スプライトの割りこみを可能にする。

●画面上のスコア表示〈行520〜550〉

行530〜540　表示場所を用意し、スコアを表示する。

●残り人数の表示〈行560〜600〉

行580〜590　表示場所を用意し、残り人数を表示する。

●ゲームオーバーの処理〈行610〜640〉

行620　画面をテキスト・モードにして、「GAME OVER（ゲームオーバー）」の文字とスコアを赤で表示する。

行630　終了画面をしばらく見せて、プログラムの実行を終わる。

●スプライト・パターンのデータ〈行650〜730〉

行660〜670　レシーバーのスプライト・パターンのデータ。

行680〜690　爆弾のスプライト・パターンのデータ。

行700〜710　大金のスプライト・パターンのデータ。

行720〜730　人のスプライト・パターンのデータ。

使用変数

DDH　落下物のスピード

ER　人を受けそこなった数

K　人を受けそこなった(＝0)か、そうでない(＝1)かを判断する

MX　レシーバーのスプライト・パターンのＸ座標

RX、RY　落下物（爆弾、大金、人）のスプライト・パターンのＸ座標、Ｙ座標

SW　スプライト・パターンが表示されている(＝1)か、そうでない(＝0)かを判断する

```
99 'オタスケ ゲーム
100 SCREEN 2,2:COLOR 15,14,4:KEY OFF
110 DEFINT A-Z:OPEN "grp:" AS #1
120 MX=100:MY=159:DMX=10:DH=10:SC=0:MH=20:ER=0:NC=20:K=0:DIM RX(3),RY(3),SW(3)
130 FOR I=0 TO 3
140     S$=""
150     FOR J=1 TO 32
160         READ A$:S$=S$+CHR$(VAL("&h"+A$))
170     NEXT J
180     SPRITE$(I)=S$
190 NEXT I
200 FOR I=1 TO 3:RY(I)=17:SW(I)=0:NEXT I
210 LINE (0,175)-(255,191),12,BF
220 PSET (10,0):COLOR 0:PRINT #1,USING "SCORE #####      ノコリ ## ニン";SC,NC
230 ON SPRITE GOSUB 420:SPRITE ON
240 ' メイン ショリ
250 A=STICK(0):IF A=3 THEN MX=MX+DMX ELSE IF A=7 THEN MX=MX-DMX
260 IF MX>255 THEN MX=MX-DMX
270 IF MX<0 THEN MX=MX+DMX
280 PUT SPRITE 0,(MX,MY),13,0
290 ' ラッカブツ ノ イドウ
300 FOR I=1 TO 3
310     IF SW(I)=0 THEN RX(I)=RND(-TIME)*200+1:SW(I)=1
320     IF I=1 THEN DDH=8 ELSE IF I=2 THEN DDH=5 ELSE IF I=3 THEN DDH=10
330     IF SW(I)=0 THEN RY(I)=RY(I)+DH ELSE RY(I)=RY(I)+DDH
340     IF RY(I)>170 THEN 370
350     PUT SPRITE I,(RX(I),RY(I)),I+(I-1)*3,I
360     RY(I)=RY(I)+DDH
370     IF RY(I)>170 THEN PUT SPRITE I,(RX(I),209)
380 NEXT I
390 FOR I=1 TO 3:IF RY(I)=>209 AND SW(I)=1 THEN SW(I)=0:RY(I)=17:NEXT I
400 IF K=1 AND RY(3)=17 THEN K=0:GOTO 410 ELSE IF K=0 AND RY(3)=17 THEN ER=ER+1
```

```
     :GOSUB 570:IF ER=>MH THEN 610
410 GOTO 240
420 ' ﾊﾝﾀﾞﾝ
430 SPRITE OFF
440 FOR P=1 TO 3:IF RY(P)<144 THEN 500:NEXT P
450 FOR P=1 TO 3
460     IF ((RX(P)>MX-16 AND RX(P)<MX+16) AND (RY(P)=>143 )) THEN 480
470 NEXT P
480 IF P=1 THEN PLAY "O5GFEDC":SC=SC-500:GOTO 610 ELSE IF P=2 THEN PLAY "A"
     :SC=SC+200:SW(2)=0 ELSE IF P=3 THEN PLAY "G":SC=SC+100:SW(3)=0:K=1
490 GOSUB 520:PUT SPRITE P,(RX(P)+150,209),0 ,P:RX(P)=RND(-TIME)*200+1:RY(P)=209
   :GOTO 500
500 SPRITE ON
510 RETURN
520 ' ｽｺｱ
530 LINE( 58,0)-( 98,16),14,BF
540 PRESET (58,0):COLOR 0:PRINT #1,USING "#####";SC
550 RETURN
560 ' ﾋﾄﾉ ﾉｺﾘｽｳ
570 NC=MH-ER
580 LINE (166,0)-(188,16),14,BF
590 PRESET (172,0):COLOR 0:PRINT #1,USING "##";NC
600 RETURN
610 ' ｹﾞｰﾑ ｵｰﾊﾞｰ
620 CLS:SCREEN 1,1:COLOR 6:LOCATE 5,10:PRINT "G A M E  O V E R":LOCATE 7,12
   :PRINT USING "SCORE ##### ﾃﾝ";SC
630 FOR I=1 TO 3000:NEXT I:CLS:END
640 WATE 1000:CLS:END
650 ' ﾃﾞｰﾀ
660 DATA 00,c0,c1,63,33,19,0f,07,03,03,03,06,0c,78,70,00
670 DATA 00,03,83,c6,cc,98,f0,e0,c0,c0,c0,60,30,1e,0e,00
680 DATA 00,00,00,00,00,06,0f,1f,3f,7f,7f,3e,1d,0f,06,00
690 DATA 02,10,05,08,12,20,c0,80,c0,e0,c0,c0,00,00,00,00
```

```
700 DATA 00,76,37,1F,0F,07,3D,78,75,F5,F8,7D,75,38,1D,0F
710 DATA 00,6E,EC,F8,F0,E0,BC,1E,AE,BF,1F,AE,AE,1C,B8,F0
720 DATA 03,01,00,00,00,38,69,cb,cf,07,1e,1d,1c,00,01,01
730 DATA 00,80,c0,40,40,c0,fc,e6,c3,81,00,e0,20,60,c0,80
```

キーボード練習

一時期はやった“モグラたたき”。どの穴から顔を出すのかわからず、しかも、出たと思ったらすぐに引っこんでしまう……。ちょっとしたタイミングと、カンがモノをいうゲームでした。いまは、おかし大作戦や足し算ゲームなど、モグラたたきの弟ぶんのようなゲームがいろいろと出まわっています。

このキーボード練習は、モグラたたきの要領でキー操作がうまくなるという、うれしいゲーム。何度もくり返しているうちに、キーボードをいちいち見なくても、操作できるようになるはずです。

①開始
F5/F10
⬇
② ? 何かキーを
⬇ 押す
③コース選択
1/2/3
⬇
④ゲーム
↑ ～ →
⬇
⑤終了

遊び方

◉コースの選択

実行すると、「キーボード練習」の文字とともに、「START HIT ANY KEY（スタート ヒット エニー キー）」と表示されるから、どれでもよいから一つキーを押す。

次にコースの選択画面があらわれ、1～3のどれにするかを聞いてくる。上級になるほど出題の時間が短いから、キー操作に不慣れなら、とりあえず1を選択しよう。

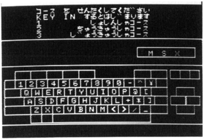
◉ゲームの進め方

コース選択の数字を押したら、しばらくして画面のキーのどれかが白くぬりつぶされる。これが問題だから、実際のキーボードから同じキーを探して、すばやくそれを押そう。

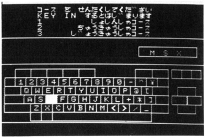
押しまちがえば失敗だし、まごまごしていると、次の問題（キー）に移る。



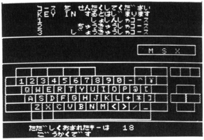
①最初の画面



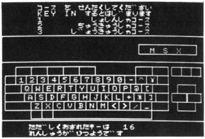
②コース選択の画面



③出題



④18点以上の場合のスコア画面



⑤17点以下の場合のスコア画面

出題数は20問で、これはどのコースを選んでも同じ。18問以上正解なら「合格です」と表示されるが、そこまでいかないと、「練習が必要です」とパソコンにいわれてしまうぞ。

ゲームのコツと注意

◎正解なら音楽が流れ、不正解なら、ブーと短い音が出る。どちらの場合も、すぐに次のキーに目を移そう。とくに不正解のときは、続けて正解のキーを押してもダメだから、早くあきらめて次の問題を待つほうがよい。もたもたしていると、次の問題も時間オーバーになる。

◎コース選択の数字を押すと、しばらくして最初の問題が出る。この間、なんの合図もないので、画面から目を離さないようにしよう。



ゲームの改造

◘問題表示の時間を変える

問題のキー表示時間は、プログラムの行460〜480の「SP＝」に続く数字で設定されている。つまり、次のようになる。

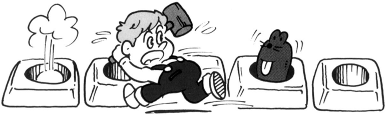
➜コース1→SP＝300

➜コース2→SP＝200

➜コース3→SP＝100

この数字が大きいほど問題表示の時間が長いから、自分のレベルに合わせて数字を変えればよい。

たとえば、コース1でもむずかしすぎるという人は、行460の「300」を、500や600に変える。逆に、コース3でもラクラクという人は、行480の「100」を80や50に減らしてみよう。



プログラムの解説

▶初期設定と最初の画面表示 <行100〜180>

行100　使用する画面を高解像度グラフィック・モードにして、色を設定する。

行110　乱数の発生を実行のたびに変わるようにし、グラフィック画面に文字が表示できるようにする。

行120〜160　最初の画面を表示する。

行170〜180　何かキーが押されるまで待ち、押されたら行180へ飛んで画面をクリアし、配列を宣言する。

▶キーボードの表示 <行190〜380>

行190　行710へ飛んで、配列に文字をセットする。

行200〜350　キーボードの図を描く。

行360　「MSX」の表示をする。

行370　行670からキーの座標のデータをA（X座標）、B（Y座標）に読みこむ。

行380　行750へ飛んで、キーの文字を表示する。

◯コースの選択と画面の表示〈行390〜480〉

行390〜430　コースの選択のための文字を表示する。

行440　コース（1〜3）が選択されるまで待つ。

行450　コースの選択からゲームの開始まで、少し時間をおく。

行460〜480　選択されたコースにより、キーを表示する時間を変える。

◯ゲームの処理〈行490〜620〉

行490　出題（キー表示）を20回くり返すループの開始。

行500〜520　乱数を発生させ、それによって問題のキーを白くぬる。

行530〜590　問題のキーを、コースによって決められた時間だけ表示し、その間に正解かどうかをチェックするループ。正解のときは音楽を鳴らし、正解数に1をプラスする。不正解のときは「レ」の音を1音だけ鳴らす。時間内にどのキーも押されなければ、次の処理に移る。

行600〜620　キーを元に戻し、出題を20回くり返してループを終わる。

◯スコアの判定と表示〈行630〜650〉

行630〜640　正解数を表示し、それが17点より大きいかどうかによって、判定の表示を変える。

行650　スコア画面をしばらく表示し、プログラムの実行を終わる。

◯各種のデータとサブルーチン〈行660〜760〉

行670〜680　キーの座標データ。

行690　キーに描く文字のデータ。

行700〜730　キーの文字を読みこむサブルーチン。

行740〜760　キーの文字を表示するサブルーチン。

使用変数

A､B　問題のキーをぬりつぶしたり、元に戻したりするための位置を決める。AがX座標、BがY座標

E$　キーの文字を入れる

G　スコア（正解数）

R　出題のために発生させた乱数を入れる

SE$　コース選択の数字（1〜3）を入れる。

SP　出題て、キーがぬりつぶされている時間

```
99 'ｷｰﾎﾞｰﾄﾞ ﾚﾝｼｭｳ
100 SCREEN 2:COLOR 15,4,7
110 K=RND(-TIME):OPEN "GRP:" AS #1
120 PSET (70,30):PRINT #1,"*****************"
130 PSET (70,40):PRINT #1,"* ｷｰﾎﾞｰﾄﾞ ﾚﾝｼｭｳ *"
140 PSET (70,50):PRINT #1,"*****************"
150 PSET (70,100):PRINT #1,"ﾁｬﾝｽ ﾊ ｲﾁﾄﾞｷﾘ!!"
160 PSET (50,180):PRINT #1,"<START   HIT ANY KEY>"
170 A$=INKEY$:IF A$="" THEN 170 ELSE 180
180 CLS:DIM E$(48),A(48),B(48)
190 GOSUB 710
200 LINE (5,45)-(250,162),15,B:LINE (170,57)-(240,70),15,B
210 LINE (5,81)-(250,81),15:LINE (23,88)-(118,100),15,B
220 LINE (166,88)-(205,100),15,B
230 FOR F=1 TO 3 :LINE (10,100+12*F)-(205,100+12*F),15:NEXT F
240 LINE (10,100)-(205,148),15,B:LINE (42,148)-(185,160),15,B
250 FOR F=1 TO 4 :LINE (23+19*F,88)-(23+19*F,100),15:NEXT F
260 LINE (186,88)-(186,100),15
270 FOR F=1 TO 14 :LINE (10+13*F,100)-(10+13*F,112),15:NEXT F
280 FOR F=1 TO 13 :LINE (17+13*F,112)-(17+13*F,124),15:NEXT F
290 FOR F=1 TO 13 :LINE (21+13*F,124)-(21+13*F,136),15:NEXT F
300 FOR F=1 TO 12 :LINE (29+13*F,136)-(29+13*F,148),15:NEXT F
310 FOR F=1 TO 2 :LINE (42+13*F,148)-(42+13*F,160),15:NEXT F
320 LINE (209,88)-(245,100),15,B
330 FOR F=1 TO 2 :LINE (209+12*F,88)-(209+12*F,100),15:NEXT F
340 LINE (216,112)-(238,124),15,B :LINE (209,124)-(227,136),15,B
350 LINE (228,124)-(245,136),15,B :LINE (216,136)-(238,148),15,B
360 PSET  (176,62) :PRINT #1," M S X "
370 RESTORE 670:FOR F=1 TO 48 :READ A(F),B(F):NEXT F
380 GOSUB 750
390 PSET (50,4)  :PRINT#1, "コース を せんたくしてください"
400 PSET (50,12) :PRINT#1, "KEY IN するとはじまります"
```

```
410 PSET (50,20) :PRINT#1, "1         しょしんしゃコース"
420 PSET (50,28) :PRINT#1, "2       ちゅうきゅうしゃコース"
430 PSET (50,36) :PRINT#1, "3     じ゛ょうきゅうしゃコース"
440 SE$=INKEY$:IF SE$="1" OR SE$="2" OR SE$="3" THEN 450 ELSE 440
450 FOR F=1 TO 1000 :NEXT F
460   IF SE$="1" THEN SP=300:GOTO 490
470   IF SE$="2" THEN SP=200:GOTO 490
480   IF SE$="3" THEN SP=100
490   FOR Q=1 TO 20
500     R=INT(RND(1)*46+1)
510     LINE(A(R)-4,B(R)-2)-(A(R)+7,B(R)+8),4,BF
520     LINE(A(R)-4,B(R)-2)-(A(R)+7,B(R)+8),15,BF
530     FOR I=1 TO SP
540       IN$=INKEY$:IF IN$="" THEN 590
550       IF L<>0 THEN 590
560       IF E$(R)<>IN$ THEN 570 ELSE 580
570       PLAY "D":L=1
580       IF L<>0 THEN 590 ELSE PLAY "L3205BAGFEDC":G=G+1:L=1
590     NEXT I
600     LINE(A(R)-4,B(R)-2)-(A(R)+7,B(R)+8),4,BF
610     PSET (A(R),B(R)):PRINT #1,E$(R)
620   L=O:NEXT Q
630   PSET (40,168): PRINT#1, "たた゛しくおされたキ-は ";G
640   IF G>17 THEN PSET (50,177): PRINT#1, "こ゛うかくて゛す" ELSE PSET (40,177):PRINT#1
, "れんしゅうか゛ひつようて゛す"
650 FOR F=1 TO 2000:NEXT F:END
660 '  "DATA DIVISION"
670 DATA 28,103,41,103,54,103,67,103,80,103,93,103,106,103,119,103,132,103,145,1
03,158,103,171,103,184,103,35,115,48,115,61,115,74,115,87,115,100,115,113,115,12
6,115,139,115,152,115,165,115,178,115,39,127,52,127,65,127,78,127,91,127,104,127
,117,127
680 DATA 130,127,143,127,156,127,169,127,182,127,47,139,60,139,73,139,86,139,99,
```

```
139,112,139,125,139,138,139,151,139,164,139,177,139
690 DATA 1,2,3,4,5,6,7,8,9,0,-,^,¥,Q,W,E,R,T,Y,U,I,O,P,@,[,A,S,D,F,G,H,J,K,L,+,*
,],Z,X,C,V,B,N,M,<,>,/,_
700 '  "READ  SECTION"
710 RESTORE 690
720 FOR F=1 TO 48 :READ E$(F):NEXT F
730 RETURN
740 '   "PRINT  SECTION"
750 FOR F=1 TO 48 :PSET (A(F),B(F)),4:COLOR15,4:PRINT #1,E$(F):NEXT F
760 RETURN
```

ザ・ビンゴ

「みなさん、25の数字がバラバラに並んだシートを持ちましたか？　私がこれから次々に数字を選んでいきますから、その数字に印をしてください。縦、横、斜めのいずれでも、五つのマス目がどこか１列そろったら、大きな声でビンゴ！　と叫んでください」

いま、パーティーなどで人気のあるビンゴ。今回、キミのお相手をするのはコンピュータのコンタ君です。コンタ君とキミが交互に数字をいい合って、早くそろったほうが勝ち。キミが勝ったときは、大声で叫ぼう。せ〜の、「ビンゴ！」

①開始

F5/F10

↓

②ゲーム
コンピュータ選択

↓

1 〜 25
＋
RETURN

↓

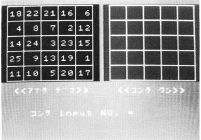
③終了

遊び方

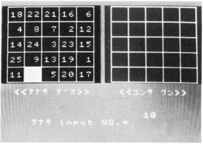
実行すると、キミのパネルが左、コンタ君のパネルが右に表示される。そのまま少し待つと、コンタ君が数字を選び、キミのパネルのその数字が黄色でぬりつぶされる。

そうなれば、次はキミが数字を選ぶ番だ。１〜25のうち、数字を一つ選んでキーインし、必ず RETURN キーを押す。そうすると、その数字がコンタ君のパネルのどこかに表示される。

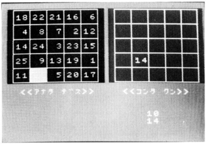
こうして交互に数字を選び、早く１列そろったほうが勝ちになる。



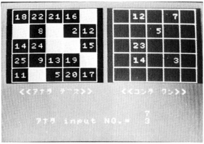
①最初の画面



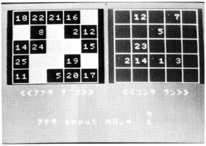
②コンタ君が数字を選ぶ



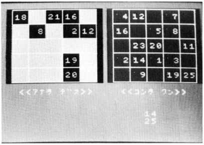
③1～25のどれかをキーイン



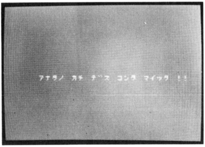
④ゲーム途中



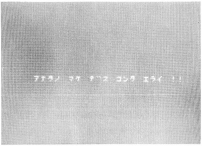
⑤あと1枚でビンゴ



⑥列がそろった



⑦アナタの勝ち



⑧アナタの負け

ゲームのコツと注意

◎カンと運だけがモノをいうゲームだから、コツはない。コンタ君のパネルがそろわないように、数字を選んでいくだけだ。

◎自分が選んだ数字でパネルをそろわせたかったら、ゲームのやり方を変えて、コンタ君側のパネルを早くそろえたほうを勝ちとすればよい。

◎数字の1～25以外や、すでにキーインした数字を押すと、ブザーが鳴る。そのときは、もう一度キーインしなおせばよい。

プログラムの解説

●初期設定〈行100〜130〉

行100　使用する画面を高解像度グラフィック・モードにし、色を決め、すべての変数を整数型に宣言する。

行110〜130　配列変数を宣言し、グラフィック画面に文字が表示できるようにして、変数 CTR を初期化する。

●ゲーム処理の流れ〈行140〜260〉

行150〜160　左側のパネルの表示。

行170〜180　右側のパネルの表示。

行190　左側のパネルに、1〜25の数をセットする。

行200　右側のパネルに、1〜25の数をセットする。

行210〜220　パネルが開かれているかどうかを調べるため、チェック用の配列に初期値をセットする。

行230　コンピュータ側が数を選ぶ処理を呼び出し、実行する。

行240　開かれたパネルが1列に並んだかどうかをチェックする。

行250　人間側が数を選び、その結果、開かれたパネルが1列に並んだかどうかをチェックする。

行260　以上の処理をして、行230に戻る。

●パネルの表示〈行270〜350〉

行150と行170で変数の値を変えることにより、左と右に色の違うパネルが表示される。

行280　四角形の表示。

行290〜310　パネルの縦線を描くループ。

行320〜340　パネルの横線を描くループ。

●人間側が数を選ぶ処理〈行360〜480〉

行370　キーインされた数字が入る変数B＄を、何もない状態にしておく。

行380　人間側への入力指示を表示する。

行390　文字の入力を待ち、入力されたら次の行へ進む。

行400　RETURN キーが押されたかどうかを判定し、押されていなければキーインされた文字をB＄に加える。押されていれば、行420へ飛ぶ。

行410　キーインされた文字が3桁(けた)を超えたかどうかを判定し、超えていれば次の

行へ進む。超えていなければ、行380に戻って文字の入力を待つ。

行420　キーインされた文字が、数字の1～25の範囲にあるかどうかをチェックする。これ以外であれば、ブザーを鳴らし、行380に戻って文字の入力を待つ。

行430　キーインされた文字が、すでに入力されたものかどうかをチェックする。入力されていなければ、配列L1の該当する場所を0にする。入力されていれば、ブザーを鳴らし、行380に戻って文字の入力を待つ。

配列L1には1～25の場所が設定されていて、数が入るとその場所に0がセットされる。たとえば10が入力されると、配列L1の10番めに0が入る。

行440　入力指示を消して、キーインされた数字を表示する場所を用意する。

行450～460　キーインされた数字を、パネルに表示する。コンタ君側のパネルの数はL2で設定されていて、L2のたとえば1番めには、数字の1が入っている場所がセットされている。その場所をパネルの座標に変換して、表示する。

行470　入力されたパネルの該当場所に、1をセットしておく。

▶コンピュータ側が数を選ぶ処理〈行490～580〉

行500～510　コンピュータ側の入力指示を表示し、音を出す。

行520　1～25の乱数を一つ発生させる。

行530　発生した数が、すでに選ばれたものかどうかをチェックする。選ばれていなければ、配列L1に0をセットして、次の行へ進む。選ばれていれば、行520に戻って、再び乱数を発生させる。

行540　選ばれた数を表示する。

行550～560　選ばれた数のパネルを、明るい黄色でぬりつぶす。しくみは、人間側（行450～460）と同じ。

行570　入力されたパネルの該当場所に、1をセットしておく。

▶パネルに数をセットする〈行590～660〉

行600　セットのための、ループの開始。

行610　2次元の配列変数L1の中の位置を、1次元にしてそれぞれ頭からかぞえて何番めに該当するかを計算し、その数をCTRに入れる。

行620～630　1～25の乱数を一つ発生させ、その数がすでに発生したかどうかをチェックする。発生していれば行620に戻り、再び乱数を発生させる。発生していなければ、パネルにセットして、L2に入れる。

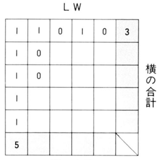
行640　パネルがコンピュータ側の場合、行660へ飛ぶ（数字を表示しない）。

行650～660　パネルの数字を表示し、処理を終わる。

▶配列の初期化〈行670〜680〉

行680　数をパネルにセットするためと、パネルが開かれたかどうかをチェックするため、配列に初期値をセットする。

▶パネルの横と縦の列のチェック〈行690〜750〉



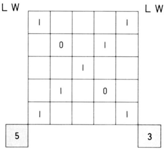
縦の合計

行700〜730　パネルを横と縦で1列ずつチェックするループ。変数LWは、開かれている場合が1、開かれていない場合が0だから、各列を合計し、その数が5であればその列がすべて開かれたことになる。

行740　横の列のどこかすべてが開かれていれば(LW＝5)、行800へ飛んで終了の処理をする。開かれていなければ、変数LWの値をすべて0に戻しておく。

行750　縦の列について、行740と同じ処理をする。

▶パネルの斜めの列のチェック〈行760〜790〉



斜めの合計

行770　左上から右下に並ぶ斜めの1列について、LWの値をチェックする。それが5であれば、すべて開かれたとして、行800の終了処理へ飛ぶ。5でなければ、LWの値を0に戻す。

行790　右上から左下に並ぶ斜めの1列について、行770と同じ処理をする。

▶終了の処理〈行800〜840〉

行800〜810　プレイヤーを判定し、勝ったのがコンピュータなら、行820へ飛ぶ。そうでなければ、音楽を鳴らし、メッセージを表示する。

行820　コンピュータが勝った場合の音楽を鳴らす(人間が勝った場合とは、メロディが違う)。

行830〜840　音楽を鳴らしている間の時間をかせぎ、メッセージを表示する。しばらく表示したあと、プログラムの実行を終わる。

使用変数

B$　キーインされた文字を入れておく

HH　パネルの色

L1　選ばれた数字が２回使われないよう、チェックするための２次元配列。1が人間側、2がコンピュータ側

L2　選ばれた数字がパネルの何番めに入っているかをチェックするための２次元配列。１が人間側、２がコンピュータ側

LU　パネルの数字を入れる配列変数。5×5のものが2枚あり、1枚め（画面左）が人間側、２枚め（画面右）がコンピュータ側

LW　パネルが開かれている（＝１）か、開かれていない（＝０）かのどちらかが入る。列がそろったかどうかの判断用

W1　コンピュータ側の乱数を入れる

W2　人間側が入力した内容を、数値に変換して入れる

WK　パネルにセットするための乱数を入れる

WW　パネルの横線を描くときに基準となるX座標

WX$　入力される文字を１文字ずつ入れる

X1　パネルに数字を表示するためのX座標

XX　プレイヤーの判定

Y1　パネルに数字を表示するためのY座標

YY　パネルの縦線を描くときに基準となるX座標



```
99 '    VINGO game
100 COLOR 15,4,1:SCREEN 2:DEFINT A-Z
110 DIM LU(2,5,5),L1(2,25),L2(2,25),LW(2,6,6)
120 OPEN"grp:" AS #1
130 CTR=0
140 '*    main                                *
150 YY=0:HH=1:WW=20:GOSUB 280
160 PSET(30,120),4:PRINT#1,"<<ｱﾅﾀ ﾃﾞｽ>>"
170 YY=120:HH=14:WW=140:GOSUB 280
180 PSET(150,120),4:PRINT#1,"<<ｺﾝﾀ ｸﾝ>>"
190 XX=1:GOSUB 680:H= 4  :W=1:GOSUB 600
200 XX=2:GOSUB 680:H=125 :W=15:SS=1:GOSUB 600
210 XX=1:GOSUB 680
220 XX=2:GOSUB 680
230 XX=1:GOSUB 500
240 XX=1:GOSUB 700
250 XX=2:GOSUB 370
260 GOTO 230
270 '*          box                           *
280 LINE(13+YY,9 )-(127+YY,111),HH,BF
290 FOR I=20 TO 120 STEP 20
300   LINE(I+YY,10)-(I+YY,110),15
310 NEXT I
320 FOR I=10 TO 110 STEP 20
330   LINE(WW,I)-(WW+100,I),15
340 NEXT I
350 RETURN
360 '*    data input  ｱﾅﾀ                     *
370 B$=""
380 PSET(50,160),5:PRINT#1,"ｱﾅﾀ input NO.="
390 WX$=INKEY$ :PSET(170+SW*8,160),4:PRINT #1,B$:IF WX$="" THEN GOTO 380
400 IF ASC(WX$)<>13 THEN B$=B$+WX$ ELSE GOTO 420
```

```
410 IF LEN(B$)>3  THEN SW=0:LINE(180,160)-(200,170),4,BF:GOTO 420 ELSE SW=1:GOTO
 380
420 W2=VAL(B$):IF (W2=0 OR W2>25)  THEN B$="":LINE(170,160)-(200,170),4,BF:BEEP:
GOTO 380
430 IF L1(XX,W2)<>0 THEN L1(XX,W2)=0:GOTO 440 ELSE B$="":LINE(170,160)-(200,170)
,4,BF:BEEP:GOTO 380
440 LINE(50,160)-(200,170),4,BF
450 X1=L2(XX,W2)MOD 5:IF X1=0 THEN X1=5 :YX=-1 ELSE YX=0
460 Y1=L2(XX,W2)/5+1+YX:PSET(20*X1+125,20*Y1-3),14:PRINT#1,USING"##";W2
470 LW(XX,X1,Y1)=1
480 XX=2:GOSUB 700:RETURN
490 '*    data input  ｺﾝﾀ                    *
500 LINE(170,150)-(250,158),4,BF
510 PSET(50,150),5:PRINT#1,"ｺﾝﾀ input NO. =":FOR I=1 TO 1000:NEXT I:PLAY"D"
520 W1=RND(-TIME)*25+1:IF (W1=0 OR W1>25) THEN GOTO 520
530 IF L1(XX,W1)=0 THEN GOTO 520 ELSE IF L1(XX,W1)=W1 THEN  L1(XX,W1)=0:GOTO 540
 ELSE 520
540 LINE(50,148)-(200,158),4,BF:PSET(170,150),4:PRINT#1,W1
550 X1=L2(XX,W1)MOD 5:IF X1=0 THEN X1=5 :YX=-1 ELSE YX=0
560 Y1=L2(XX,W1)/5+1+YX:LINE(20*X1+1 ,20*Y1-9)-(20*X1+19,20*Y1+9),11,BF
570 LW(XX,X1,Y1)=1
580 RETURN
590 '*    ｹｲｻﾝ                        *
600 FOR I=1 TO 5:FOR J=1 TO 5
610   CTR=I-5+J*5
620   WK=RND(-TIME)*25+1
630   IF L1(XX,WK)=0 THEN GOTO 620 ELSE LU(XX,I,J)=WK:L2(XX,WK )=CTR:L1(XX,WK)=0
640   IF SS=1 THEN GOTO 660
650   PSET(I*20+H,J*20-3),W :PRINT#1,USING"##";LU(XX,I,J);
660 NEXT J:NEXT I:RETURN
670 '*    flag1 init                  *
680 FOR I=1 TO 25:L1(XX,I)=I:NEXT I:RETURN
```

```
690 '*    ﾖｺﾄ ﾀﾃﾉ ﾊﾝﾀﾞﾝ                *
700 FOR I=1 TO 5:FOR J=1 TO 5
710   LW(XX,6,I)=LW(XX,6,I)+LW(XX,J,I)
720   LW(XX,I,6)=LW(XX,I,6)+LW(XX,I,J)
730 NEXT J:NEXT I
740 IF (LW(XX,6,1)=5 OR LW(XX,6,2)=5 OR LW(XX,6,3)=5 OR LW(XX,6,4)=5 OR LW(XX,6,
5)=5) THEN GOTO 800 ELSE FOR I=1 TO 5:LW(XX,6,I)=0 :NEXT I
750 IF (LW(XX,1,6)=5 OR LW(XX,2,6)=5 OR LW(XX,3,6)=5 OR LW(XX,4,6)=5 OR LW(XX,5,
6)=5) THEN GOTO 800 ELSE FOR I=1 TO 5:LW(XX,I,6)=0 :NEXT I
760 '*    ﾅﾅﾒﾉ ﾊﾝﾀﾞﾝ 1  L-R         *
770 FOR I=1 TO 5:LW(XX,6,6)=LW(XX,6,6)+LW(XX,I,I):NEXT I:IF LW(XX,6,6)=5 THEN GO
TO 800 ELSE LW(XX,6,6)=0
780 '*    ﾅﾅﾒﾉ ﾊﾝﾀﾞﾝ 2  R-L         *
790 FOR I=1 TO 5:LW(XX,6,6)=LW(XX,6,6)+LW(XX,I,6-I):NEXT I:IF LW(XX,6,6)=5 THEN
GOTO 800 ELSE I.W(XX,6,6)=0:RETURN
800 IF XX=2 THEN GOTO 820 ELSE  PLAY"o4g2e4.f8g2c2d8e8f8g8f4e4d1e8f8g8a8g4r16g4o
5c2o4g2f4e4d4.c8r16c1"
810 FOR I=1 TO 10000:NEXT I:CLS:PSET(50,100),4:PRINT#1,"ｱﾅﾀﾉ ｶﾁ ﾃﾞｽ ｺﾝﾀ ﾏｲｯﾀ !!"
:FOR I=1 TO 10000:NEXT I:GOTO 840
820 PLAY"o4g4o5c4.r16c8r16c4e4d4.c8d4e4c4.r16c8e4g4a2.r16a4g4.e8r16e4c4d4.c8d4e4
c4.o4a8r16a4g4o5c2."
830 FOR I=1 TO 10000:NEXT I:CLS:PSET(50,100),4:PRINT#1,"ｱﾅﾀﾉ ﾏｹ ﾃﾞｽ ｺﾝﾀ ｴﾗｲ !!":
FOR I=1 TO 10000:NEXT I
840 END
```

着陸ゲーム

——昔、人類ハ次々ニ宇宙へ飛ビ出シ、第二ノ地球トナル惑星ヲ探シ求メタ。ソシテイマ、ツイニソノ星ヲ見ツケタガ——

このアンドロメダ星系第99番惑星は、人類が第二の地球としてねらいをつけた星だ。この星を偵察(ていさつ)するため、われわれの仲間を乗せた「宇宙船艦カルチャー号」は、いまようやく着陸のチャンスを得た。しかし、この星にはいん石の落下が多く、着陸は簡単ではない。人類すべての希望を乗せたカルチャー号は、はたして着陸に成功するか、それはひとえにキミの操縦手腕にかかっている。

①開始
F5/F10
↓
②何かキーを押す
□
↓
③ゲーム
↑ ← → ↓
↓
④成功／失敗
↓
⑤終了

遊び方

●ゲームの進め方

実行すると、「START(スタート)＝ANY(エニー)」という表示が出る。心と指の準備ができたら、どれでもよいから一つキーを押そう。

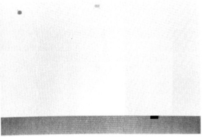
ゲームがスタートすると、着陸船といん石があらわれて動きはじめる。画面の下の灰色の帯が地面で、その上の１か所に、青い台が出る。これが、キミのめざす着陸ポイントだ。

カーソル移動キーで宇宙船を動かして、いん石をかわしながら、うまく着陸させよう。成功すると「SUCCESS(サクセス)(成功)」と、かかった時間が表示される。

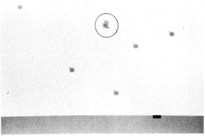
いん石に当たったり、着陸ポイントではない地面に降りてしまったら、その時点でゲームオーバー。「FAILURE(フエイリヤ)（失敗)」と表示され、ゲーム終了となる。



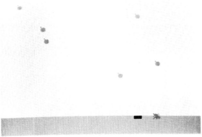
①最初の画面



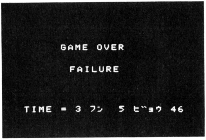
②ゲームスタート



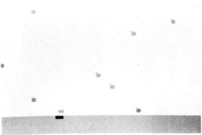
③いん石とぶつかる



④ほかの場所に着陸



⑤失敗の画面



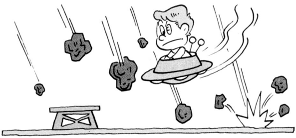
⑥着陸成功



⑦成功の画面

● 着陸船の操縦

カーソル移動キーの←を押すと、宇宙船は左下、→を押すと右下、↑↓なら上下に動く。何も押さないと、宇宙船はゆっくり真下に落ちていく。



ゲームのコツと注意

◎ゲームがスタートしたら、いん石の動きに注意して、すぐに宇宙船を移動させよう。スタートと同時にドカン、ということも多いからだ。

◎操縦のコツは、⬆キーをうまく使うこと。上に動いていん石をやりすごせば、あとは楽にいくことが多い。

ゲームの改造

◘いん石、宇宙船の動くスピードを変える

プログラムの行150で、「IS＝４」がいん石のスピード、「SS＝８」が宇宙船のスピードとして設定されている。この数字のほうを、いろいろ変えてみよう。数値を大きくするほど、動くスピードが速くなる。

プログラムの解説

➡初期設定〈行100〜120〉

行100 使用する画面を高解像度グラフィック・モードにし、色を決める。

行110 すべての変数を整数型に宣言し､乱数の発生が実行のたびに変わるようにする。また、配列を宣言する。

行120 グラフィック画面に文字を表示できるようにし､ブザー音の大きさと長さを決める。

➡ゲーム処理の流れ〈行130〜240〉

行130〜140 ゲームのタイトルを表示し、何かキーが押されるのを待つ。

行150 画面を消し、各変数に初期値をセットする。

行160 スプライト･パターンに使う変数を何もない状態にセットして、IS＄にいん石をあらわす●マークを入れる。

行170 爆発音のデータを読みこむ。

行180〜200 スプライト・パターンのデータを読みこみ、着陸船、着陸場所、爆発したときの形をそれぞれつくる。また、LINE(ライン)文でゲーム画面を描き、乱数によって地面のどこか１か所に着陸場所を表示する。

行210 画面の最上段に、着陸船を表示する。

行220 スプライトの衝突による割りこみの飛び先を行620にして、割りこみを可能にする。

行230　いん石の発生、着陸船の操作、いん石の移動の各処理を呼び出してゲームを実行する。

▶いん石発生の処理〈行250〜300〉

行260〜270　いん石の数が画面に10以上ある場合、また、前のいん石との間隔が12以下のときは、いん石を発生させる処理を行わない。

行280〜290　いん石表示のX座標と落下方向を乱数で求め、いん石を表示する。また、画面に表示しているいん石の数を、1プラスする。

▶着陸船の操作〈行310〜400〉

行320　変数DSに、カーソル移動キーの押された方向を入れる。

行330〜380　カーソル移動キーが押された方向によって、着陸船を移動し、表示する。

行390　着陸船の位置（Y座標）が地面とぶつかる範囲内かどうかを調べ、ぶつかっていれば失敗処理（行720）のあと、終了処理（行620）を呼び出す。

▶いん石の移動と消去〈行410〜600〉

行420　表示されているいん石が0なら、この処理を実行しない。そうでなければ、消去するいん石の数DIを0にセットしておく。

行430　いん石の数だけ以下の処理をくり返す、ループの開始。

行440　移動のために、いん石を消す。

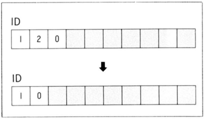
行450〜460　いん石が移動する方向によって、移動する座標を計算する。

行470〜480　いん石が画面の左右両端から外に出たか、あるいは地面の位置に接触しているかどうかを調べ、外に出たり、地面に落ちたいん石を消すために、いん石の番号をセットする。

行490　新しい位置に、いん石を表示する。

行500〜510　X軸方向とY軸方向で、いん石が着陸船と衝突しているかどうかをチェックする。衝突していれば、Eに1をセットする。

行520　ループの終わり。



行530　いん石が着陸船と衝突していれば、失敗処理（行720）のあと、終了処理（行620）を呼び出す。

行540　消去するいん石がないときは、この処理を終わる。

行550〜580　消去するいん石のデータ

(X座標、Y座標、落下方向)を消すため、配列を前に詰める。たとえば、落下方向を示す配列IDで、表示されているいん石の数が3、消去するのが2番めのいん石の場合、前ページの図のようになる。

行590　表示されているいん石の数を、消去したぶんだけマイナスする。

●ゲーム終了の処理〈行610〜700〉

行620　スプライトの衝突による割りこみを禁止する。

行630　ゲーム成功のとき、所要時間を表示するために行790へ飛ぶ。

行640〜700　終了表示のための画面を設定し、「GAME OVER」の表示を出す。また、着陸に成功したとき、失敗したときによって、メッセージと所要時間を表示する。終了画面をしばらく表示し、プログラムの実行を終わる。

●衝突処理〈行710〜770〉

行720　スプライトの衝突による割りこみを禁止する。

行730　所要時間を計算し、着陸船の位置に爆発のスプライト・パターンを重ねる。

行740〜760　爆発音を出す。

●所要時間の計算〈行780〜790〉

行790　所要時間を分、秒単位で計算する。

●DATA文と音楽の演奏〈行800〜870〉

行810　スプライト・パターンのためのデータ。

行820　爆発音をつくる**SOUND**文のためのデータ。

行830〜860　成功のとき、音楽を鳴らす。

使用変数

- **BK$**　爆発のスプライト・パターン
- **CH**　消去するいん石の番号
- **DI**　消去するいん石の数
- **DR$**　着陸場所のスプライト・パターン
- **E**　成功(=0)か失敗(=1)かの判定
- **I**　いん石といん石との間隔
- **ID**　いん石の落下方向(0のとき垂直、1のとき左、2のとき右)
- **IS**　いん石のスピード
- **NI**　画面に表示されているいん石の数
- **S1、S2**　爆発音をつくるためのデータを入れる
- **SP$**　着陸船のスプライト・パターン
- **SS**　着陸船のスピード
- **XI、YI**　いん石のX座標、Y座標を入れる配列
- **XS、YS**　着陸船のX座標、Y座標

```
99 '  ﾁｬｸﾘｸ ｹﾞｰﾑ
100 SCREEN 2,2:COLOR 1,15:CLS
110 DEFINT A-Z:T=RND(-TIME):DIM S1(13),S2(13)
120 OPEN "GRP:" FOR OUTPUT AS #1 : SET BEEP 3,4
130 PSET (80,72):PRINT #1,"ﾁｬｸﾘｸ ｹﾞｰﾑ":PSET (100,90):PRINT #1,"START=ANY"
140 Z$=INKEY$:IF Z$="" THEN GOTO 140
150 CLS:E=0:NI=0:I=12:IS=8:SS=4
160 SP$="":DR$="":BK$="":IS$=CHR$(133)
170 RESTORE 820:FOR C=0 TO 13:READ S1(C),S2(C):NEXT C
180 RESTORE 810:FOR C=1 TO 8:READ A$,B$,C$:SP$=SP$+CHR$(VAL("&H"+A$)):DR$=DR$+CH
R$(VAL("&H"+B$)):BK$=BK$+CHR$(VAL("&H"+C$)):NEXT C
190 XC=INT(RND(1)*235)+1:YC=162:LINE (1,1)-(255,YC),15,BF:LINE (1,YC+1)-(255,189
),14,BF:PUT SPRITE 2,(XC+10,YC),4
200 SPRITE$(0)=BK$:SPRITE$(1)=SP$:SPRITE$(2)=DR$
210 XS=120:YS=5:PUT SPRITE 1,(XS,YS),4:TIME=0
220 ON SPRITE GOSUB 620:SPRITE ON
230 GOSUB 260:GOSUB 320:GOSUB 420
240 GOTO 230
250 ' ｲﾝｾｷ ﾉ ﾊｯｾｲ
260 IF NI>=10 THEN GOTO 300
270 IF NI<>0 THEN IF YI(NI)<I THEN GOTO 300
280 NI=NI+1:YI(NI)=1:XI(NI)=INT(RND(1)*245)+1:ID(NI)=INT(RND(1)*3)
290 LINE (XI(NI)*.9,YI(NI)*.9)-(XI(NI)*.9+8,YI(NI)*.9+8),15,BF
300 RETURN
310 ' ﾁｬｸﾘｸｾﾝ ｿｳｻ
320 DS=STICK(0)
330 IF DS=1 THEN IF YS<=1 THEN GOTO 380 ELSE YS=YS-SS:GOTO 380
340 IF DS=5 THEN YS=YS+SS:GOTO 380
350 IF DS=3 THEN IF XS>=247 THEN GOTO 380 ELSE XS=XS+SS:GOTO 370
360 IF DS=7 THEN IF XS<=1 THEN GOTO 380 ELSE XS=XS-SS
370 YS=YS+(SS/2)+(SS MOD 2)
380 PUT SPRITE 1,(XS,YS),4
```

```
390 IF YS>YC-5 THEN E=1:GOSUB 720:GOSUB 620
400 RETURN
410 ' ｲﾝｾｷ ﾉ ｲﾄﾞｳ
420 IF NI<1 THEN GOTO 600 ELSE DI=0
430 FOR C3=1 TO NI
440     LINE(XI(C3),YI(C3))-STEP(8,8),15,BF
450     IF ID(C3)=1 THEN XI(C3)=XI(C3)-IS ELSE IF ID(C3)=2 THEN XI(C3)=XI(C3)+IS
460     YI(C3)=YI(C3)+IS
470     IF XI(C3)>=1 AND XI(C3)<=245 THEN GOTO 480 ELSE DI=DI+1:CH(DI)=C3-DI+1:GO
TO 500
480     IF YI(C3)>=1 AND YI(C3)<=YC-8 THEN GOTO 490 ELSE DI=DI+1:CH(DI)=C3-DI+1:G
OTO 500
490     PSET (XI(C3),YI(C3)),1:COLOR 9:PRINT #1,IS$
500     IF XS+8>XI(C3) AND XS-8<XI(C3) THEN GOTO 510 ELSE GOTO 520
510     IF YS+8>YI(C3) AND YS-8<YI(C3) THEN C3=NI:E=1
520 NEXT C3
530 IF E=1 THEN  GOSUB 720:GOSUB 620
540 IF DI=0 THEN GOTO 600
550 FOR C=1 TO DI
560     IF CH(C)=NI THEN GOTO 580
570     FOR C2=CH(C) TO NI-1:XI(C2)=XI(C2+1):YI(C2)=YI(C2+1):ID(C2)=ID(C2+1):NEXT
 C2
580 NEXT C
590 NI=NI-DI
600 RETURN
610 ' GAME OVER
620 SPRITE OFF
630 IF E=0 THEN GOSUB 790
640 FOR C=1 TO 2000:NEXT C:SCREEN 2,2:COLOR 15,1:CLS
650 PSET (90,72):PRINT #1,"GAME OVER"
660 IF E<>0 THEN PSET (100,100):PRINT #1,"FAILURE":GOTO 690
670 PSET (100,90):PRINT #1,"SUCCESS"
680 GOSUB 830
```

```
690 PSET (50,150):PRINT #1,USING "TIME =## ﾌﾝ ## ﾋﾞｮｳ ##";H,B,M
700 FOR C=0 TO 3000:NEXT C:CLOSE #1:END
710 ' ｼｮｳﾄﾂ ｼｮﾘ
720 SPRITE OFF
730 GOSUB 790:PUT SPRITE 0,(XS,YS),8
740 FOR C=0 TO 13:SOUND S1(C),S2(C):NEXT C
750 FOR C=0 TO 1000:NEXT C
760 FOR C=0 TO 13:SOUND C,0:NEXT C
770 RETURN
780 ' ｼﾞｶﾝ ｹｲｻﾝ
790 H=TIME/3600:B=(TIME-H*3600)/60:M=(TIME-H*3600) MOD 60:RETURN
800 ' DATEﾌﾞﾝ
810 DATA 42,FF,92,3D,FF,54,7E,FF,38,FF,00,FF,7E,00,1C,3D,00,2A,42,00,49,81,00,88
820 DATA 0,112,1,13,2,3,3,10,4,54,5,10,6,23,7,198,8,31,9,31,10,31,12,75,11,189,1
3,8
830 PLAY "O5V10L8AR16L16AR16L16AR16L2A","O5V10L8FR16L16FR16L16FR16L2F","O5V10L8C
R16L16CR16L16CR16L2C"
840 PLAY "O5V10L4CCC","O5V10L4GO4FG","O5V10L4EO4AE"
850 PLAY "O5V10L1E","O6V10L1C","O5V10L1G"
860 PLAY "O4V10L16GR16L16GR16L16G","O4V10L16ER16L16ER16L16E","O4V10L16CR16L16CR1
6L16C"
870 RETURN
```

くろひげ恋うらない

いつだって、だれかに恋をしていたい！　だれもが思うことですが、恋に悩みはつきもの。そんなとき、くろひげ君を相手に恋の行方をうらなってみましょう。

25枚のパネルがすべて開けば、成就まちがいなし。くろひげ君が、にっこり笑って祝福してくれます。ところが、へんなパネルを開くと、恐ろしいドクロがあらわれ、キミの恋もそこまで。さあ、勇気を出してアタックしてください。

①開始
F5/F10
↓
②ゲーム
1 ～ 25＋ RETURN
↓
③成功／失敗
↓
④終了

遊び方

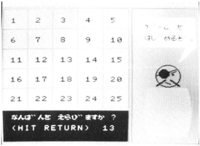
●ゲームの進め方

実行すると、軽快な音楽とともに、25枚のパネルとくろひげが出てくる。くろひげがゲーム開始の合図を出し、画面の左下には「何番を選ぶか？」というメッセージが出る。

ここから、ゲームのスタートだ。注意するのは、25枚のパネルのうち、必ずどこか1枚にドクロが隠れていること。運悪くそのパネルを選ぶと、不気味な音楽をバックにドクロがあらわれ、ゲームオーバーになってしまう。



①最初の画面



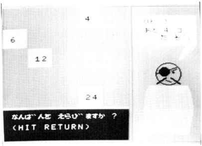
②パネル番号をキーイン



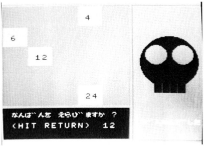
③ＯＫ、だいじょうぶ



④キーインミス



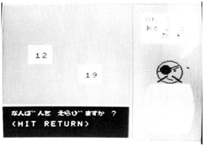
⑤あと4枚



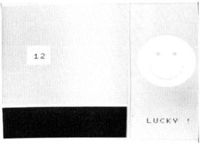
⑥残念、ドクロがあらわれた



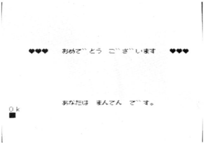
⑦スコアの表示



⑧よし、あと2枚



⑨くろひげが笑顔に



⑩満点の画面

そこでキミは、ドクロのパネルを予測して、だいじょうぶだと思われるパネルの番号を押し、RETURN キーを押す。ドクロのパネル以外なら、そのパネルが赤でぬりつぶされ、くろひげが「あと×個」というメッセージを出す。

●満点の場合

こうして24枚までパネルを開くことに成功したら、くろひげが自動的に笑顔に変わり、キミの恋を祝福してくれる。つまり、成功だ。

●ゲームオーバーの場合とうらないの結果

途中で運悪く、ドクロのパネルを選んでしまったら、くろひげがドクロに変わり、ゲームオーバーとなってスコア（それまでに開いたパネル数）が表示される。これが、キミの恋うらないの結果だ。スコアの高い人はまだまだチャンスがあるから、下表の診断を参考にしてほしい。



スコア	診断
0点	可能性ゼロ。一生、恋のチャンスなし
1～5点	頼れるものはカネだけ。カネの力で押しの一手だ
6～10点	ファッションかスポーツで、相手の目をこちらにひきつけよう
11～15点	古典的だが、相手の気をひくラブレター作戦で
16～20点	電話攻勢でアタック、アタック、アタック！
21点	相手の親兄弟、ペットをまず手なずけよう
22点	さりげない食事の誘いで、意外にうまくいくかも
23点	あともう少し。相手はキミの強引な誘いを待っている

ゲームのコツと注意

◎パネル番号をキーインしたら、そのあと必ず RETURN キーを押すこと。押さないと、いつまでたっても画面が変わらない。

◎パネルにないキーや、一度押した数字を押してしまったときは、「input（インプット） key（キー）

error！」という表示が出て、キーイン画面が元に戻る。そこで、もう一度押しなおせばよい。

◎いったん入れてしまった番号を変更したいときは、**RETURN** キーを押す前に、何かどれでもほかのキーを押せばよい。やはり「input key error！」が出て、押しなおすことができる。

プログラムの解説

▶初期設定〈行100〜140〉

行100　使用する画面を高解像度グラフィック・モードにして、ファンクション・キーの表示を消し、変数を整数型に宣言する。

行110　ゲーム開始の音楽を鳴らす。

行120　グラフィック画面に文字を表示できるようにし、画面の色を決める。

行130　変数 RS に 1 〜25のうち、一つの数をランダムにセットする。これを、実行のたびに変わるようにする。

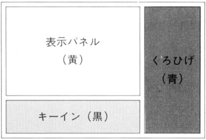
行140　配列変数 AA に、1 〜25の数値をセットする。

▶ゲーム画面と絵を描く〈行150〜370〉

行150〜200　画面を数字表示パネル、キーイン画面、くろひげ表示画面の三つに、図のように分ける。

行210〜280　くろひげと、タルの絵を描く。

行290〜370　表示パネルに 5 × 5 のマス目をつくり、1 〜25の番号を表示する。



▶ゲームの開始〈行380〜670〉

行380　変数 AT に残りパネル数25をセットし、**COLOR** 文で赤を設定する。

行390〜420　ゲーム開始のメッセージ欄をつくり、メッセージを表示する。

行430　ゲームの開始音を鳴らす。

行440〜450　キーイン待ちのメッセージを表示する。

行460〜520　キーインされた内容が、1 〜25の番号と **RETURN** キーかどうかをチェックする。それ以外なら行930へ飛んで入力ミスの表示をし、行660へ飛ぶ。キーインが正しければ、選択された番号を IK $ に入れ、それを数値型変数に変換する。また、何も押されなければ、押されるまで待つ。

行530　選択された番号がすでに使われたものかどうかをチェックし、そうなら行930へ飛んで入力ミスの表示をして、行660へ飛ぶ。そうでなければ、AAに0をセットして、一度しか使えないようにする。

行540　選択された番号と、行130でセットされた番号（ドクロ）が同じかどうかを調べ、そうなら行810へ飛んで終了処理をする。

行550〜650　ドクロの番号以外の数がキーインされたときは、音楽の合図とともに、そのパネルを赤でぬりつぶし、残り数(変数AT)を計算してそのメッセージを表示する。

行660〜670　画面を黒に戻す。残りのパネル数が1かどうかをチェックし、1でなければ行440に戻ってキーイン待ちのメッセージを表示する。

▶満点の場合の処理〈行680〜790〉

行690　合図の音楽を鳴らす。

行700〜750　くろひげを消し、ニコニコマークを描いて「LUCKY（ラッキー）！」と表示する。

行770〜790　画面をテキスト・モードにして、満点のメッセージを表示する。

▶ゲームオーバーの処理〈行800〜910〉

行810〜820　合図の音楽を鳴らす。

行830〜880　くろひげを消し、ドクロを描いて、残念でしたと表示する。

行890〜910　画面をテキスト・モードにして、「GAME OVER（ゲーム オーバー）」の文字とスコアを表示する。

▶入力ミスの処理〈行920〜950〉

行930〜940　合図音とともに、ミスの表示を出す。

行950　しばらく表示して、キーイン待ちに戻る。

使用変数

AA	1〜25の番号をセットする
AT	パネル（番号）の残り数
IK	選択番号（数値）
IK$	選択番号（文字）
PX	パネル番号を消すときのX座標
PY	パネル番号を消すときのY座標
RS	ドクロマークが隠れている番号
SX	パネル番号を消すときのX軸方向の位置
SY	パネル番号を消すときのY軸方向の位置
X	パネルを表示するときの起点となるX座標
Y	パネルを表示するときの起点となるY座標

```
99 ' ｸﾛﾋｹﾞ
100 SCREEN 2: KEY OFF: DEFINT A-Z
110 PLAY"L3205BAGFEDC"
120 OPEN "grp:" AS #1: COLOR 15,5,7
130 RS=INT(RND(-TIME)*25+1)
140 DIM AA(25): FOR I=1 TO 25:AA(I)=I:NEXT I
150 LINE (0,0)-(255,191),5,BF
160 LINE (8,3)-(160,140),8,B
170 LINE (9,4)-(159,139),10,BF
180 LINE (8,143)-(160,188),8,B
190 LINE (9,144)-(159,187),1,BF
200 LINE (164,3)-(252,188),8,B
210 CIRCLE (208,145),57,14,2.5,3.8,1.9: CIRCLE (208,145),57,14,5.65,.65,1.9
220 LINE (184,111)-(232,111),14: LINE (184,179)-(232,179),14: PAINT (218,112),14
230 CIRCLE (208,95),15,1: CIRCLE (203,92),6,1: PAINT (203,92),1
240 LINE (193,95)-(218,86),1
250 LINE (205,100)-(194,108),1: LINE (205,101)-(194,109),1
260 LINE (211,100)-(222,108),1: LINE (211,101)-(222,109),1
270 LINE (205,104)-(211,106),1
280 LINE (212,91)-(218,88),1: CIRCLE (214,93),1,1:PAINT (214,93),1
290 FOR I=1 TO 4
300   LINE (I*30+9,3)-(I*30+9,140),8
310   LINE (8,I*27+4)-(160,I*27+4),8
320 NEXT I
330 X=17:Y=16:K=1: COLOR 5
340 FOR I=0 TO 4:FOR J=0 TO 4
350   PSET (X+30*J-6,Y+27*I),10
360   PRINT #1,AA(K): K=K+1
370 NEXT J:NEXT I
380 AT=25: COLOR 8
390 LINE (174,15)-(242,55),15,BF
400 LINE (230,55)-(222,75),15: LINE (242,55)-(222,75),15: PAINT (231,56),15
```

```
410 PSET (180,25),15:PRINT #1,"ゲーム を"
420 PSET (180,40),15:PRINT #1,"はじめるぞ。"
430 PLAY"l106R4C"
440 PSET (20,150),1:COLOR 15: PRINT #1,"なんばんを えらびますか ?"
450 PSET (20,165),1: PRINT #1,"(HIT RETURN)"
460 PSET (130,165),1: A$=INKEY$: IF A$="" THEN 460 ELSE PRINT #1,A$
470 IF ASC(A$)<48 OR ASC(A$)>57 THEN GOSUB 930:GOTO 660
480 PSET (138,165),1: B$=INKEY$: IF B$="" THEN 480 ELSE PRINT #1,B$
490 IF ASC(B$) <> 13 THEN IK$=A$+B$ ELSE IK$=A$: GOTO 520
500 IF ASC(B$)<48 OR ASC(B$)>57 THEN GOSUB 930:GOTO 660
510 PSET (96,165),1: C$=INKEY$: IF C$="" THEN 510 ELSE IF ASC(C$) <> 13 THEN GOS
UB 930: GOTO 660
520 IK=VAL(IK$): IF IK<1 OR IK>25 THEN GOSUB 930:GOTO 660
530 IF AA(IK)=0 THEN GOSUB 930: GOTO 660 ELSE AA(IK)=0
540 IF IK=RS THEN 810
550 PLAY"L3205BAGFEDC"
560 SX=IK MOD 5: IF SX=0 THEN SX=5
570 SY=INT((IK-1)/5)
580 PX=(SX-1)*30+10: PY=SY*27+5
590 PAINT (PX,PY),8
600 IF AT=25 THEN LINE (174,15)-(242,55),15,BF
610 AT=AT-1: COLOR 8
620 PSET (180,20),15: PRINT #1,"OK !"
630 LINE (203,32)-(233,40),15,BF
640 PSET (180,32),15: PRINT #1,"あと";AT;"コ"
650 PSET (200,42): PRINT #1,"だぞ。"
660 LINE (9,144)-(159,187),1,BF
670 IF AT<>1 THEN GOTO 440
680 '    ***  おわり (まんてん)  ***
690 FOR I=1 TO 3: PLAY "l64o5cdefg": NEXT I:PLAY "l1o6r4c"
700 LINE (165,4)-(251,187),5,BF
710 CIRCLE (208,80),35,11: PAINT (208,56),11
720 CIRCLE (208,85),20,4,3.14,6.28
```

```
730 CIRCLE (198,70),2,4: PAINT (198,70),4
740 CIRCLE (218,70),2,4: PAINT (218,70),4
750 PSET (185,160):PRINT #1,"LUCKY !"
760 FOR I=1 TO 2000: NEXT I
770 SCREEN 1
780 LOCATE 3,8: PRINT "♥♥♥  おめでとう ございます  ♥♥♥"
790 LOCATE 8,17: PRINT "あなたは まんてん です。":END
800 '   ***  おわり (ざんねん)  ***
810 PLAY"T1701404CR32C8.R32C16R32C4E-8.D16R32"
820 PLAY"1404D8.C16R32C08.03B16R3204C4.R4"
830 LINE (165,4)-(251,187),5,BF
840 CIRCLE (208,80),35,1: PAINT (208,56),1: LINE (190,80)-(226,122),1,BF
850 CIRCLE (192,70),13,5:CIRCLE (224,70),13,5: PAINT (192,58),5: PAINT (224,58),
5
860 FOR I=1 TO 3: LINE (190+9*I,110)-(190+9*I,122),5: NEXT I
870 PSET (175,160): PRINT #1,"ざんねんでした"
880 FOR I=1 TO 2000: NEXT I
890 SCREEN 1: PLAY "O3L1E"
900 LOCATE 4,8: PRINT "***  GAME  OVER  ***"
910 LOCATE 3,17: PRINT "あなたの とくてんは";25-AT;"てん です。": END
920 '   ***  ERROR ROUTINE  ***
930 PLAY "12o6c"
940 COLOR 8:PSET (30,175),1:PRINT #1,"input key error !"
950 FOR I=1 TO 500 :NEXT I:RETURN
```

音当てクイズ

「ソ・ド・ドド・ミ・レ・ドレ」これはなんのメロディでしょう。そう、よく耳にする「蛍の光」です。では、これらの音をバラバラにして、一つの音だけで出してみたら、いったいいくつの音を当てることができるでしょうか。

音楽をメロディで覚えるのは簡単ですが、単音や和音をそれだけでポンと当てるには、ちょっとした耳の訓練が必要。まずはパソコンで、自分の音感を試してみてください。

①開始
F5/F10
↓
②何かキーを押す
?
↓
③音符の選択
? + Y
↓
④オクターブの選択
1 2 + ?
↓
⑤単音、和音の選択
1 2
↓
⑥ゲーム
何かキーを押す
?
↓
答えをキーイン
↓
判定
↓
⑦ゲーム続行？
Y ／ ?
↓
⑧終了



遊び方

●音符、オクターブの選択

実行すると「START＝ANY」(スタート＝エニー)という文字が出てくる。何かキーを押すとスタートしますよ、という意味だから、どれかキーを一つ押す。

すると、パソコンが音の長さを聞いてくる。同時にゲームの基準となる「ド」の音が全音符で鳴らされる。その音でOKならYをキーイン。もっと短い音を基準にしたいのなら、何か別のキーを一つ押す。１回押すたびに、音の長さが２分

音符、4分音符、……と32分音符まで変化する。自分が決めた音符のところでY をキーインして、音の長さが決定。

次に、オクターブ選択の画面になり、音符選択で決定した「ド」の音が、同じオクターブ（高さ）で鳴らされる。もっと高い音を基準にしたいのなら1、低い音なら2をキーイン。どちらも1回押すごとに、音の高さが1オクターブずつ変化する。この場合は、1 2以外のキーをどれか一つ押して高さが決定する。

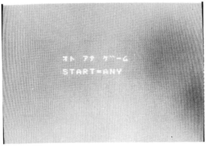
●単音と和音の選択

音符とオクターブが決定したら、その「ド」の音を基準にして、単音を当てるか和音を当てるかの選択画面になり、鍵盤があらわれる。単音なら鍵盤のC～Bの音と1～5の半音で、合計12のうちから1音が問題として鳴らされる。

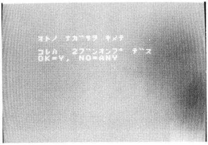
和音なら、「ドミソ」「ドファラ」「シレソ」のうちから、一つが問題として鳴らされる。1か2をキーインして、ゲームのスタートだ。

●ゲームの進め方

単音、和音のどちらの場合でも、まず「準備OK？」の画面になる。心ならぬ耳の準備ができたら、どれでもよいから一つキーを押そう。同時に、問題の音が鳴らされる。



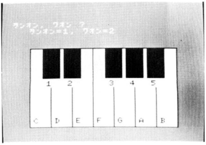
①最初の画面



②音符の選択



③オクターブの選択



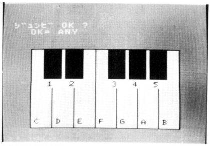
④単音、和音の選択

次に、「もう一度聞く？」という画面に変わるから、確信できるまで何度でも聞けばよい。

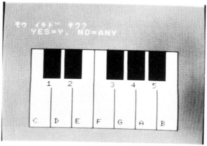
解答のしかたは、単音と和音とで違うから、それぞれ画面の指示にしたがうこと。正解なら「セイカイ」、不正解ならそのメッセージと正解が示される。

◉ゲームの続行と終了

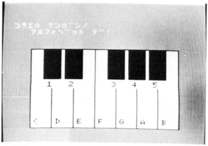
１回のゲームが終わると、「ゲーム続行？」の画面になる。続行なら[Y]キーインで、音符選択の画面に戻る。そこで、再び音符選択→オクターブ選択→単音、和音の選択、と進めていく。



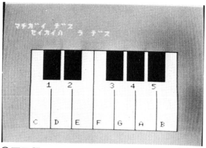
⑤準備ＯＫ？



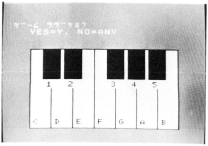
⑥もう一度聞く？



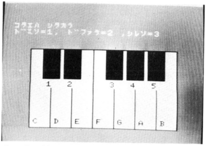
⑦単音の答え入力



⑧不正解



⑨ゲーム続行？



⑩和音の答え入力

こうして、何回かゲームをくり返し、終わりたいときは、「ゲーム続行？」の画面でⓎ以外のキーを押す。出題回数のうち、正解が何パーセントだったかによって、次のようなメッセージが表示される。

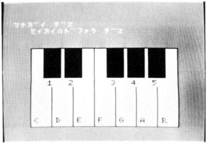
➡20%以下→ヤーイ、オンチ！

➡40%以下→ザンネン　デシタ

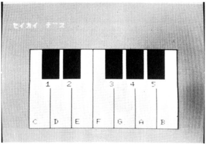
➡60%以下→オツカレサマ　デシタ

➡80%以下→オミゴト　デシタ

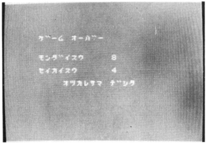
➡81%以上→アンタガ　タイショウ！



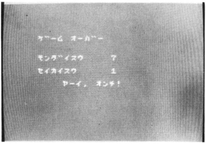
⑪不正解



⑫正解



⑬得点とメッセージ1



⑭得点とメッセージ2

ゲームのコツと注意

◎音符選択から単音と和音の選択まで、キー操作が少しややこしい。画面の指示をよく見て、慎重にキーインすること。

◎聞き慣れてくると、半音が出るぶんだけ単音のほうがむずかしい。また、音符は短くなるほど、オクターブは上や下にいくほどむずかしい。正解するごとに音符を一つ短くするなど、ゲームをくふうしてみよう。

◎10ゲームで1回戦、などと決めて何人かで得点を競ってみよう。

ゲームの改造

◘和音の数を増やす

以下の各行を変えることによって、和音をいくつでも増やすことができる。

①行310の「W＝1　TO　3」で和音の数をセットしている。「TO」のあとの3を、増やした合計数に変える。

②行560で、解答のために和音を表示している。「シレソ＝3」のあとに、増やしたぶんだけ追加する。

③行810で、どの和音を出題するか、1～3をでたらめに出すことによって決めている。「RND(1)＊3」の3を、増やした合計数に変える。

④行960が和音のデータになっている。これを参考にして、増やしたぶんだけデータを追加する。



プログラムの解説

▶初期設定〈行100～150〉

行100　使用する画面を高解像度グラフィック・モードにし、色を決め、画面を消去する。

行110　すべての変数を整数型に宣言し、配列変数を用意し、乱数の発生が実行のたびに変わるようにする。

行120　グラフィック画面に文字が表示できるようにし、**SET BEEP** 命令でブザー音の音色と大きさを決める。

行130　タイトルと「START＝ANY」を表示する。

行140　問題数、正答数の変数に、初期値0をセットする。

行150　行910を呼び出し、何かキーが押されるのを待つ。

◗音の設定〈行160〜290〉

行160〜220　音の長さ（音符）を決めるための画面表示と、キーインにしたがって音符を変え、その長さで「ド」を出す処理。

行230〜290　音の高さ（オクターブ）を決めるための画面表示と、キーインにしたがってオクターブを変え、その高さで「ド」を出す処理

◗音の設定と鍵盤の表示〈行300〜410〉

行310　和音のコードを読みこみ、変数に入れる。

行320　単音のコードを読みこみ、変数に入れる。

行330〜410　鍵盤の表示。

◗ゲームの処理〈行420〜610〉

行430〜450　単音、和音の選択表示と、キーインを待つ。

行460〜490　1か2がキーインされたら、「準備OK」の表示を出し、何かキーが押されるのを待つ。

行500　1が押された場合は単音を出す処理（行750）を呼び出し、そうでない場合（2をキーイン）は和音を出す処理（行800）を実行する。

行510〜530　もう一度聞くかどうかの選択を待ち、Yがキーインされたら行470に戻る。

行540　単音が選ばれた場合、解答キーインを指示するメッセージを表示する。

行550〜560　和音が選ばれた場合、解答キーインを指示するメッセージを表示する。

行570〜580　解答のキーインを待ち、正解かどうかを判定する処理を呼び出す。メッセージをしばらく表示して、次の処理に移る。

行590〜610　ゲームを続行するかどうかのメッセージを表示し、キーインを待つ。続けるなら行160に戻り、続けないなら次の処理に移る。

◗ゲームの終了処理〈行620〜730〉

行630　正解率をパーセントで計算する。

行640〜670　正解率に応じて、メッセージを用意する。

行680〜730　「ゲームオーバー」と問題数、正解数、メッセージをしばらく表示し、プログラムの実行を終わる。

●単音を出す処理〈行740〜780〉

行750　もう一度聞く選択がなければ（Yキーイン）、1～12の乱数を発生させる。

行760〜780　発生させた乱数に応じた音を出す。行750によって、もう一度聞く選択があった場合は、同じ音が出る。

●和音を出す処理〈行790〜840〉

行800　もう一度聞く選択があった場合、行830へ飛んで、同じ音が出るようにしておく。

行810〜840　乱数を発生させて和音を選び、その和音を出す。

●正誤の判定〈行850〜890〉

行860　キーインされた解答と出した音をチェックし、正解のときはブザーを鳴らし、正解の表示をする。正解数に1をプラスし、行890へ飛んで問題数にも1をプラスする。

行870〜880　不正解の処理。「ド」の音を鳴らし、不正解の表示をする。同時に、正解の音を表示する。

行890　問題数に1をプラスする。

●キーイン待ちの処理〈行900〜910〉

行910　何かキーが押されるのを待ち、押されれば次の処理に移る。

●各データ〈行920〜960〉

行930　画面に鍵盤を描くとき、黒い鍵盤をつくるためのX座標のデータ。

行940〜950　単音のコードと名称。

行960　和音のコードと名称。

使用変数

AC	正答数	TN$	音のコード
LG$	音の長さ(音符)	TS$	単音の名称
OC$	音の高さ(オクターブ)	TW$	単音か和音かの選択
OT$	出題された音のコード	WT$	和音のコード
QC	問題数	WS$	和音の名称

```
99 '  ｵﾄｱﾃ ｹﾞｰﾑ
100 SCREEN 2:COLOR15,5:CLS
110 DEFINT A-Z:DIM TN$(12):DIM TS$(12):T=RND(-TIME)
120 OPEN "GRP:" FOR OUTPUT AS #1:SET BEEP 3,4
130 PSET (80,72):PRINT #1,"ｵﾄ ｱﾃ ｹﾞｰﾑ":PSET (80,90):PRINT #1,"START=ANY"
140 QC=0:AC=0
150 GOSUB 910
160 CLS:TR=0:LN$(0)="  ｾﾞﾝｵﾝﾌﾟ":LN$(1)=" 2ﾌﾞﾝｵﾝﾌﾟ":LN$(2)=" 4ﾌﾞﾝｵﾝﾌﾟ":LN$(3)=" 8
ﾌﾞﾝｵﾝﾌﾟ":LN$(4)="16ﾌﾞﾝｵﾝﾌﾟ":LN$(5)="32ﾌﾞﾝｵﾝﾌﾟ"
170 C=0:L=1:PSET(50,45):PRINT #1,"ｵﾄﾉ ﾅｶﾞｻｦ ｷﾒﾃ"
180 LINE (72,63)-(152,71),5,BF
190 PSET (50,63):PRINT #1,"ｺﾚﾊ ";LN$(C);" ﾃﾞｽ":PSET (50,72):PRINT #1,"OK=Y, NO=A
NY"
200 G=LEN(STR$(L))-1:LG$="L"+RIGHT$(STR$(L),G):PLAY LG$+"O4C"
210 GOSUB 910 : IF A$="Y" THEN GOTO 230
220 C=C+1:L=L*2:IF C>5 THEN GOTO 170 ELSE GOTO 180
230 CLS:PSET (50,45):PRINT #1,"ｵｸﾀｰﾌﾞｦ ｷﾒﾃ":PSET (50,63):PRINT #1,"ｲﾏﾉ ｵｸﾀｰﾌﾞﾃﾞ
ﾄﾞｦ ﾀﾞｼﾃｲﾏｽ"
240 PSET (50,81):PRINT #1,"ﾀｶｸ=1, ﾋｸｸ=2, ｹｯﾃｲ=ANY":OC=4
250 PSET (80,135),5:COLOR15:LINE(80,135)-(250,142),5,BF:OC$="O"+RIGHT$(STR$(OC),
1):PLAY LG$+OC$+"C"
260 AZ$=INKEY$:IF AZ$="" THEN 260
270 IF AZ$="1" AND OC=8 THEN PSET (80,135):COLOR15:PRINT #1,"ｺﾚｲｼﾞｮｳ ﾀｶｸ ﾃﾞｷﾏｾﾝ"
:GOTO 260
280 IF AZ$="2" AND OC=1 THEN PSET (80,135):COLOR15:PRINT #1,"ｺﾚｲｼﾞｮｳ ﾋｸｸ ﾃﾞｷﾏｾﾝ"
:GOTO 260
290 IF AZ$="1" THEN OC=OC+1:COLOR 5:GOTO 250 ELSE IF AZ$="2" THEN OC=OC-1:COLOR
5:GOTO 250
300 '  ｵﾄﾉｾｯﾃｲ ﾄ ｹﾝﾊﾞﾝﾉﾋｮｳｼﾞ
310 RESTORE 960:FOR W=1 TO 3:READ WT$(W),WS$(W):NEXT W
320 RESTORE 940:FOR C=1 TO 12:READ TN$(C),TS$(C):NEXT C
330 RESTORE 930:X=25
```

```
340 CLS:WN=3:LINE (1,59)-(255,180),5,BF:LINE (40,63)-(215,171),15,BF
350 LINE (40,63)-(40,171),1:LINE (40,171)-(215,171),1
360 LINE (215,171)-(215,63),1:LINE (40,63)-(215,63),1
370 FOR C=1 TO 12
380   IF C<8 THEN IF C<>3 THEN LINE (40+X,63)-(40+X,104),1:LINE (40+X,117)-(40+X
,171),1 ELSE LINE (40+X,63)-(40+X,171),1 ELSE GOTO 400
390   PSET (20+X,159):COLOR1:PRINT #1,TN$(C):COLOR15:GOTO 410
400   READ XX:LINE (XX,63)-(XX+20,104),1,B:LINE (XX,63)-(XX+20,104),1,BF:PSET (X
X+5,108),15:COLOR1,15:PRINT #1,RIGHT$(STR$(C-7),1):COLOR15,5
410 X=X+25:NEXT C
420 '  ｹﾞｰﾑ ｼｮﾘ
430 LINE(20,27)-(250,50),5,BF
440 PSET (20,27):PRINT #1,"ﾀﾝｵﾝ, ﾜｵﾝ ?":PSET (40,36):PRINT #1,"ﾀﾝｵﾝ=1, ﾜｵﾝ=2  "
450 TW$=INKEY$: IF TW$="" THEN 450
460 IF TW$="1" OR TW$="2" THEN GOTO 470 ELSE GOTO 450
470 LINE(20,27)-(250,50),5,BF
480 PSET (20,27):PRINT #1,"ｼﾞｭﾝﾋﾞ OK ?":PSET (40,36):PRINT #1,"OK= ANY"
490 A$=INKEY$:IF A$="" THEN 490
500 IF TW$="1" THEN GOSUB 750 ELSE GOSUB 800
510 LINE(20,27)-(200,50),5,BF:PSET (20,27):PRINT #1,"ﾓｳ ｲﾁﾄﾞ ｷｸ?":PSET (40,36):P
RINT #1,"YES=Y, NO=ANY":AG$=""
520 AG$=INKEY$:IF AG$="" THEN 520:IF AG$="Y" THEN 470
530 IF AG$="Y" THEN 470
540 IF TW$="1" THEN LINE(20,27)-(250,50),5,BF: PSET (20,27):PRINT #1,"ｺﾀｴﾊ ｹﾝﾊﾞﾝ
ﾉ":PSET (40,36):PRINT #1,"ｱﾙﾌｧﾍﾞｯﾄ ﾃﾞ!    ":GOTO 570
550 LINE(20,27)-(250,50),5,BF
560 PSET (20,27):PRINT #1,"ｺﾀｴﾊ ｼﾀｶﾗ  ":PSET (20,36):PRINT #1,"ﾄﾞﾐｿ=1, ﾄﾞﾌｧﾗ=2 ,
ｼﾚｿ=3 "
570 AO$=INKEY$: IF AO$="" THEN 570
580 GOSUB 860:FOR C=0 TO 2000:NEXT C
590 LINE(20,27)-(250,50),5,BF
600 PSET (20,27):PRINT #1,"ｹﾞｰﾑ ﾂﾂﾞｹﾙ?":PSET (40,36):PRINT #1,"YES=Y, NO=ANY  "
610 GOSUB 910 : IF A$="Y" THEN  GOTO 160
```

```
620 '  ゲーム シュウリョウ
630 RI=(AC/QC)*100
640 IF RI<=20 THEN EV$="ヤーイ, オンチ!   ":GOTO 680
650 IF RI<=40 THEN EV$="ザンネン デシタ  ":GOTO 680
660 IF RI<=60 THEN EV$="オツカレサマ デシタ":GOTO 680
670 IF RI<=80 THEN EV$="オミゴト デシタ  " ELSE EV$="アンタガ タイショウ!"
680 CLS:COLOR 15,5:PSET (50,45):PRINT #1,"ゲーム オーバー"
690 PSET (50,72):PRINT #1,USING "モンダイスウ    ##";QC
700 PSET (50,90):PRINT #1,USING "セイカイスウ    ##";AC
710 PSET (80,108):PRINT #1,EV$
720 FOR C=0 TO 5000:NEXT C
730 CLOSE #1:END
740 '  タンオン
750 IF AG$<>"Y" THEN TN=INT(RND(1)*12)+1
760 PLAY LG$+OC$+TN$(TN)
770 IF TN>7 THEN OT$=RIGHT$(STR$(TN-7),1) ELSE OT$=TN$(TN)
780 RETURN
790 '  ワオン
800 IF AG$="Y" THEN GOTO 830
810 OT=INT(RND(1)*3)+1:OT$=RIGHT$(STR$(OT),1)
820 FOR W=1 TO 3:WN$(W)=LG$+OC$+MID$(WT$(OT),W,1):NEXT W
830 PLAY WN$(1),WN$(2),WN$(3)
840 RETURN
850 '  セイゴ ハンテイ
860 IF OT$=AO$ THEN FOR C=1 TO 2:BEEP:NEXT C:LINE(20,27)-(250,50),5,BF:PSET (20,
27):PRINT #1,"セイカイ デス   ":AC=AC+1:LINE(40,36)-(168,44),5,BF:GOTO 890
870 PLAY "L103C":LINE(20,27)-(250,50),5,BF:PSET (20,27):PRINT #1,"マチガイ デス  "
880 PSET(40,36):IF TW$="1" THEN PRINT #1,"セイカイハ";TS$(TN);" デス  " ELSE PRINT #1,
"セイカイハ"WS$(OT)" デス"
890 QC=QC+1:RETURN
900 '  キーイン
910 A$=INKEY$:IF A$="" THEN GOTO 910 ELSE RETURN
920 '  データ
```

```
930 DATA 55,80,130,155,180
940 DATA "C"," ﾄﾞ","D","  ﾚ","E","  ﾐ","F"," ﾌｧ","G","  ｿ","A","  ﾗ","B","  ｼ"
950 DATA "C+","ﾄﾞ#","D+"," ﾚ#","F+","ﾌｧ#","G+"," ｿ#","A+"," ﾗ#"
960 DATA "CEG"," ﾄﾞﾐｿ","CFA","ﾄﾞﾌｧﾗ","DGB","  ｼﾚｿ"
```

隠されたアルファベット

画面のわくの中に、アルファベットの1文字が隠されています。さて、その文字は、キミのすきな彼女（彼）のイニシャルか、それとも、にくいだれかのイニシャルか？秘密のヴェールをはぐように、黒いパネルをめくってみましょう。

何枚めのパネルを開いたところで、正解がわかるか——カンとヒラメキが勝負を決めるゲームです。



①実行
F5/F10
↓
②ゲーム方法
[1] か [2]
↓
③ CAPS LOCK
↓
④ゲーム
[1] ～ [9]
↓
[A] ～ [Z]
（くり返す）
↓
⑤アタリ／オワリ
↓
⑥終了

遊び方

◉ゲーム方法の選択

隠されたアルファベットは、大文字のA～Z。パネル数は9で、1枚開くごとに、9分の1の部分があらわれる。

実行すると、ゲーム方法の選択画面になる。パネルを自分で開いていくなら[1]、コンピュータにまかせるなら[2]をキーインする。

◉ゲームの進め方

自分で開く場合は、1～9のうちどのパネルを見たいかを決め、その数字をキーインする。コンピュータまかせの場合は、自動的にどれか1枚のパネルが開かれる。

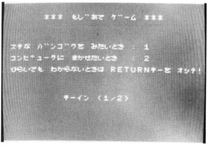
1枚開かれたところで、A～Zのどれかを予測して、英大文字をキーインする

（CAPS LOCK キーをロックしておく）。はずれた場合は、「ダメ」と表示され、続いて「アト　×　カイ」と出る。

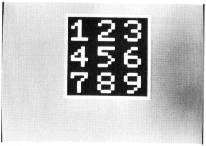
正解の場合は、パネルがすべて開かれ、「アタリ」と表示される。

はずれた場合は、次のパネルを開き、同じようにして正解を予測する。こうして、パネルを開く→予測（解答キーイン）をくり返し、何枚めのパネルで正解したかを競うゲームだ。

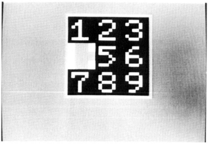
どう考えてもわからない場合は、答えないで RETURN キーを押すと、次のパネルが開かれるのを待つ処理となる。



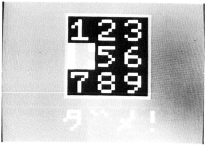
①最初の画面（ゲーム方法の選択）



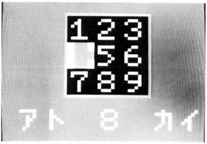
②ゲーム画面



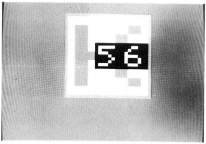
③パネルが１枚開かれる



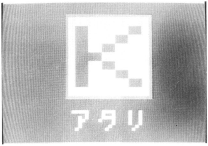
④解答をキーイン。ダメ！



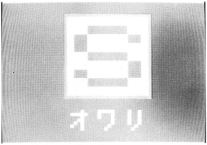
⑤アト　×　カイ



⑥あと２枚



⑦アタリ



⑧残念、オワリ

ゲームのコツと注意

◎正解を予測するとともに、どのパネルを開くかということも、このゲームの決め手になる。したがって、コンピュータまかせのほうがむずかしい。

◎解答は、必ず英大文字でキーインしないと、正解のキーを押しても、はずれとみなされる。ゲームの前に CAPS LOCK キーをロックしておこう。



プログラムの解説

初期設定とゲーム方法の選択表示〈行100〜220〉

行100　使用する画面をテキスト・モードにし、色を決め、画面を消去する。

行110　画面の表示桁(けた)数を決め、ファンクション・キーの表示を消す。

行120〜210　タイトルとゲーム方法を表示する。

行220　1か2がキーインされるのを待つ。

出題文字の設定〈行230〜350〉

行240　画面を高解像グラフィック・モードにし、画面を消去する。

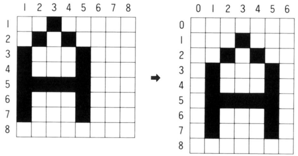
行250　すべての変数を整数型に宣言し,乱数の発生が実行のたびに変わるようにする。また、各変数に初期値0をセットする。

行260　配列変数を宣言する。

行270　グラフィック画面に文字が表示できるようにする。

行280〜290　A〜Zを乱数で発生させ、グラフィック画面に書きこむ。

行300〜350　発生した文字のパターンを読みこみ、配列変数AS（63）に入れる。具体的には、8×8のパターンで出されたデータを下図のように7×9の形に切り出し、それを1次元配列に変換して収納している。



◆パネルの表示〈行360〜460〉

行370　画面を低解像度グラフィック・モードに変える。

行380〜390　パネルと外わくをつくる。

行400〜460　パネル番号として1〜9を表示する。

◆ゲームの処理〈行470〜640〉

行480　ゲーム方法の選択で1（自分で選ぶ）がキーインされたとき、次に押されるキーの内容をY＄に入れる。選択で2がキーインされたときは、1〜9の乱数を発生させ、行510へ飛ぶ。

行490〜510　1〜9のキーが押されるのを待ち、それを数値に変換して変数Hに入れる。何も押されなかったり1〜9以外のキーが押されたときは、行470に戻る。

行520　すでに開かれているパネル番号が押されたときは、行470に戻る。

行530　開かれたパネル番号に1をセットする。また開かれたパネル数に1をプラスする。

行540〜550　パネルを開き、文字の一部を表示する処理を呼び出す。

行560　開かれたパネルが9になれば、行600へ飛んで終了処理をする。

行570〜580　解答が入力されるのを待ち、それを文字コードに変換して変数に入れる。RETURNキーだけが押されたときは、行480に戻る。

行590　解答が不正解のとき、その処理（行660）を呼び出して実行し、行480に戻って次のキーインを待つ。

行600〜630　開かれたパネルが9になれば、「オワリ」を表示して音楽を鳴らす。パネルが8枚以内で正解したときは、パネルをすべて開く処理（行950）を呼び出

し、「アタリ」を表示して別の音楽を鳴らす。

行640　終了画面をしばらく表示して、プログラムの実行を終了する。

◆不正解の場合の表示〈行650〜710〉

行660　パネルの残り枚数をWNにセットする。

行670〜680　「ダメ」としばらく表示して、音を出す。

行690〜700　パネルの残り枚数を「アト　×　カイ」と表示する。

◆文字をつくる各ブロックの色づけ〈行720〜780〉

行730〜780　文字パターンの各ブロックは、低解像度グラフィック・モードの2×3ドットで構成されている。行870で①のドットに色がぬられていて、それをもとに行730で図の②のドット、行740〜770のループで③〜⑥の各ドットをすべて同じ色にすることにより、出題文字の各ブロックが色づけされる。

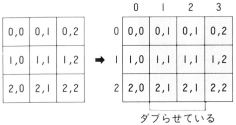
①	②
③	④
⑤	⑥

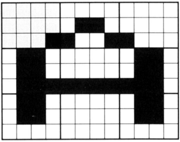
◆パネルを開く処理〈行790〜930〉

行800〜820　開くパネルの左上端の座標（S, T）と、文字パターンが入った配列のどこから表示するか（N）を計算する。

行830〜920　文字パターンの入った配列から、押されたパネル番号のパネルだけを開くループ。文字パターンは、パネル番号で分けられた9枚のパネルからなり、各パネルは、行850によって、図のように3×3ブロックを4×3ブロックに形を変えて表示するしくみになっている。たとえば文字Aは、各パネルと各ブロックが図のように色づけされる。

このループによって、1ドットごとの色が白か赤かを調べ、ドットの色づけ処理（行730）を行い、パネルを開く。





◆すべてのパネルを開く処理〈行940〜980〉

行950〜980　パネルが開かれているかどうかを調べ、開かれていなければ、パネルを開く処理を呼び出す。

使用変数

AS　文字パターンの各ドットのカラー番号を入れる配列変数

AN$　出題文字(A～Z)

H　開くパネルの番号(1～9)

P　パネルを開いたか(＝1)、開いていないか(＝0)のどちらかが入る

SN$　ゲーム方法。自分でパネルを開くか(＝1)、コンピュータにまかせるか(＝2)が入る

W$　キーインされた解答

Y$　開くパネルの番号。押されたキーの内容が入る



```
99 '  ﾓｼﾞｱﾃ
100 SCREEN 1:COLOR 15,4,7:CLS
110 WIDTH 29:KEY OFF
120 LOCATE 5,2
130 PRINT "*** もじあて ゲーム ***"
140 LOCATE 0,7
150 PRINT "スキな バンゴウを みたいとき : 1"
160 LOCATE 0,9
170 PRINT "コンピュータに まかせたいとき  : 2"
180 LOCATE 0,11
190 PRINT "ひらいても わからないときは RETURNキーを オシテ!"
200 LOCATE 8,16
210 PRINT "キーイン (1/2)"
220 SN$=INKEY$:IF SN$<>"1" AND SN$<>"2" THEN 220
230 '  ﾓｼﾞﾉ ｾｯﾃｲ
240 SCREEN 2:CLS
250 DEFINT A-Z:I=RND(-TIME):N=0:L=0:U=0
260 DIM AS(63):DIM P(9)
270 OPEN "GRP:" AS #1
280 AN$=CHR$(INT(RND(1)*26)+65)
290 PRESET (1,1):PRINT #1,AN$
300 FOR I=0 TO 8
310   FOR J=0 TO 6
320     A=POINT(J,I)
330     N=N+1:AS(N)=A
340   NEXT J
350 NEXT I
360 '  ﾊﾟﾈﾙﾉ ﾋｮｳｼﾞ
370 SCREEN 3:N=0
380 LINE (82,17)-(183,129),10,B
390 LINE (86,21)-(179,125),1,BF
400 FOR I=0 TO 2
```

```
410   FOR J=0 TO 2
420     U=U+1
430     PRESET (91+J*32,26+I*36),1
440     PRINT #1,USING"#";U
450   NEXT J
460 NEXT I
470 '   ゲーム
480 IF SN$="1" THEN Y$=INKEY$ ELSE H=INT(RND(1)*9)+1:GOTO 520
490 IF Y$="" THEN 480
500 H=VAL(Y$)
510 IF H=0 THEN 480
520 IF P(H)=1 THEN 480
530 P(H)=1:L=L+1
540 LINE (10,150)-(250,190),4,BF
550 GOSUB 800
560 IF L>8 THEN 600
570 W$=INKEY$:IF W$="" THEN 570
580 W=ASC(W$):IF W=13 THEN 480
590 IF W$<>AN$ THEN GOSUB 660:GOTO 480
600 IF L>8 THEN D$="オワリ" ELSE D$="アタリ":GOSUB 950
610 IF D$="アタリ" THEN PLAY"v12l18o5cdefgcdefgcdefl4g"
620 IF D$="オワリ" THEN PLAY"v12l10o5co4bagfedl4c"
630 PRESET (90,150):PRINT #1,D$
640 FOR K=0 TO 2000:NEXT K:END
650 '   フセイカイ
660 WN=9-L
670 PRESET (80,150):PRINT #1,"ダメ!":FOR I=1 TO 1000:NEXT I
680 PLAY"v13l4o3c#"
690 LINE (80,150)-(250,190),4,BF
700 PRESET (30,150):PRINT #1,"アト";WN;"カイ!"
710 RETURN
720 '
730 PSET STEP(4,0),C
```

```
740 FOR K=1 TO 2
750   PSET STEP(-4,4),C
760   PSET STEP(4,0),C
770 NEXT K
780 RETURN
790 '  ﾊﾟﾈﾙｦ ﾋﾗｸ
800 IF H<4 THEN N=(H-1)*2+1:S=70+N*16:T=22:GOTO 830
810 IF H<7 THEN N=(H-4)*2+22:S=70+((H-4)*2+1)*16:T=58:GOTO 830
820 IF H<10 THEN N=(H-7)*2+43:S=70+((H-7)*2+1)*16:T=94
830 FOR J=0 TO 2
840   FOR I=0 TO 3
850     IF I=2 THEN N=N-1
860     IF AS(N)<>15 THEN C=15 ELSE C=9
870     PSET (S+I*8,T+J*12),C
880     GOSUB 730
890     N=N+1
900   NEXT I
910   N=N+4
920 NEXT J
930 RETURN
940 '
950 FOR H=1 TO 9
960   IF P(H)<>1 THEN GOSUB 800
970 NEXT H
980 RETURN
```

アミダクジ

席決め、賞品の分配などものごとを決めるのによく使われるのがアミダクジ。スピードクジのように、すぐに結果が出ないところが、このクジのおもしろさであるともいえます。

このゲームは、パソコンで遊ぶアミダクジ。だれがピタリと的中させるか、大勢で集まって、一発チャンスのギャンブル気分を味わってください。

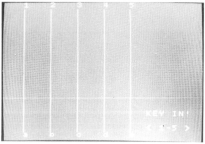
①開始
F5/F10
↓
②ゲーム
1～5
↓
③結果
↓
④終了

遊び方

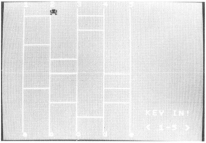
●ゲームの進め方

実行すると、1～5の数字がついた縦線と、「KEY　IN　1－5」の表示があらわれる。1～5のどれかのキーを押してくださいということだから、たとえばaにいくのはどれかと予測して、ひらめいた数字のキーを押そう。

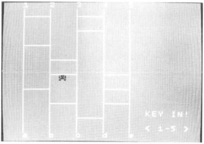
キーインすると、その数字の下にアミダ君があらわれ、同時にクジの横線が引かれる。そのまま少し待つと、アミダ君は音楽とともに、アミダクジを下にたど



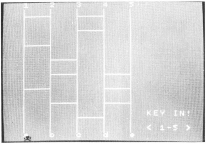
①最初の画面



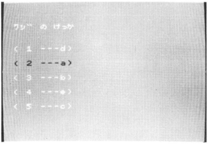
②横線とアミダ君があらわれる



③アミダ君が進む



④アルファベットに到達



⑤クジの結果

っていく。

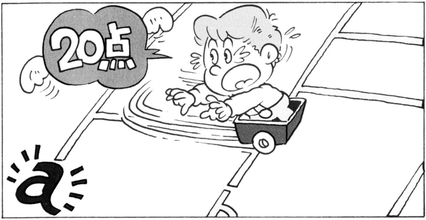
◉クジの結果

アミダ君がａ～ｅのどれかに到達すると、しばらく画面がストップし、やがて結果が表示される。キミの選んだ数字だけ色が違うので、カンが正しかったかどうかが、すぐにわかるぞ。

ゲームのコツと注意

◎ゲーム自体は単純だから、遊び方をくふうしてみよう。たとえば、ａにねらいを定め、何人かが順に数字を選ぶ。一人が数字をキーインする前に、ほかの人もａにいきそうな数字を選んでおく。ほかの人が正解なら10点、キーインした人が正解なら20点というように、何回かゲームして、得点を競えばよい。

◎上記の方法に加え、１～５のすべてのいき先を予測して、正解ならボーナスポイント100点になる、などというのもおもしろい。



プログラムの解説

▶初期設定〈行100〜220〉

行100　画面を消去し、使用する画面を高解像度グラフィック・モードにして、色を決める。

行110　配列を宣言し、乱数の発生が実行のたびに変わるようにする。

行120　グラフィック画面に文字が表示できるようにする。

行130〜220　スプライト・パターンのデータをつくり、定義する。

▶ゲーム画面の表示〈行230〜320〉

行240　音楽を鳴らす。

行250〜260　「KEY　IN」「〈1－5〉」の表示。

行280〜320　アミダの縦線と番号を表示するループ。行290で「1〜5」の数字、行300で縦線、行310で「a〜e」が表示される。

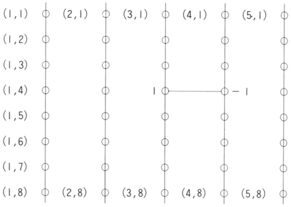
▶数字を選ぶ処理〈行330〜350〉

行340　何かキーが押されるのを待ち、押されたら、その内容を数値に変換する。

行350　数値が1〜5の範囲内であれば、次の処理に移る。それ以外なら、行340に戻る。

▶ゲームの処理〈行360〜730〉

行370〜440　乱数によってアミダの横線をつくるためのループ。2次元配列BUは図のようになっていて、各点ごとに横線が出ているかどうかをセットする。また、行380によって縦線の各間隔には横線を3本ずつ引き、行400で横線が2本続かないよう処理している。



行410〜420で、右側に横線が出ている点には1、左側に横線が出ている点には－1がセットされる。

行450　スプライト・パターンを、キーインされた縦線の上に表示する。

行460〜510　BU の情報によって、横線を表示するループ。

行520　音楽を鳴らす。

行530〜560　スプライト・パターンを、最初の横線のところまで縦に移動させるループ。横線まできたら、ほんの少しストップさせる。

行570〜630　クジの結果になる1～5とa～eの対応を調べるループ。

行640　キーインされた数字をXに、BU の各点のY軸方向の間隔（横線が引かれる段差）をRにセットする。

行650　XX にXを入れる。

行660　音楽を鳴らす。

行670〜710　BU の情報をもとに、スプライト・パターンを移動させるループ。BU が1なら、右向きの移動処理を呼び出したあと、縦の移動を行う。BU が－1なら、左向きの移動処理を呼び出したあと、縦の移動を行う。

行720　最後の、短い縦の移動処理。

行730　画面をしばらく表示したあと、結果表示処理をして、プログラムの実行を終わる。

スプライト・パターンの移動 〈行740〜920〉

行750〜810　右向きの移動。次の縦線の位置まで移動して、スプライト・パターンがある縦線の数字に1をプラスする。同時に、音楽を鳴らす。

行820〜870　左向きの移動。次の縦線の位置まで移動して、スプライト・パターンがある縦線の数字に1をマイナスする。同時に、音楽を鳴らす。

行880〜920　縦の移動。Rの値だけ、縦に移動する。

結果の表示 〈行930〜1020〉

行940〜950　スプライト・パターンを消し、結果のメッセージを表示する。

行960〜990　1～5とa～eの対応を、キーインされた数字のところだけ赤で、そのほかは緑で表示する。

行1000〜1020　音楽を鳴らし、しばらく画面を表示して処理を終わる。

使用変数

BU	それぞれの点から横線が出ているかどうかの情報を入れる
IN	選択された番号の数値
R	縦への移動距離
X	スプライト・パターンがある縦線の番号

```
99 'AMIDA
100 CLS: SCREEN 2:COLOR 15,4,13:KEY OFF
110 DIM BU(5,8):DIM XP(5):DIM A$(5,2):H=RND(-TIME)
120 OPEN "GRP:"AS #1
130 S1$=CHR$(&B00111100)
140 S2$=CHR$(&B10111101)
150 S3$=CHR$(&B10111101)
160 S4$=CHR$(&B10011001)
170 S5$=CHR$(&B11111111)
180 S6$=CHR$(&B00111100)
190 S7$=CHR$(&B11000011)
200 S8$=CHR$(&B11000011)
210 SO$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
220 SPRITE$(0)=SO$
230 ' ｹﾞｰﾑｶﾞﾒﾝﾉ ﾋｮｳｼﾞ
240 PLAY"T255E#R64E#R64E#R64E#R64GFR64FR64GR64GR64GR64F#F#F#F#"
250 PSET (180,150),4:COLOR 15:PRINT #1,"KEY IN!"
260 PSET (180,170),4:PRINT #1,"< 1-5 >":COLOR 15
270 '
280 FOR I=1 TO 5
290  PSET (I*32,0):PRINT #1,CHR$(&H30+I)
300  LINE (I*32,10)-(I*32,180)
310  PRINT  #1,CHR$(&H60+I)
320 NEXT I
330 '
340 IN$=INKEY$:IF IN$ ="" GOTO 340 ELSE IN=VAL(IN$)
350 IF IN>0 AND IN<6 THEN 360 ELSE GOTO 340
360 '
370 FOR J=1 TO 4
380  FOR C=1 TO 3
390   U=INT(RND(1)*8+1)
400   IF BU(J,U)<>0 THEN 390
```

```
410    BU(J,U)=1
420    BU(J+1,U)=-1
430   NEXT C
440  NEXT J
450 PUT SPRITE 0,(IN*32,5),3
460 FOR X=1 TO 4
470  FOR Y=1 TO 8
480   IF BU(X,Y)<>1 THEN 500
490   LINE (32*X,20*Y)-(32*(X+1),20*Y)
500  NEXT Y
510 NEXT X
520 PLAY"T255GR40GR64GR64E#FFFFR3"
530 FOR A=5 TO 20
540   PUT SPRITE 0,(IN*32,A),3
550   FOR T=1 TO 5:NEXT T
560 NEXT A
570 FOR X=1 TO 5
580  XP(X)=X:P=X
590   FOR H=1 TO 8
600    XP(X)=XP(X)+ BU(P,H)
610    IF BU(P,H)=1 THEN P=P+1 ELSE IF BU(P,H)=-1 THEN P=P-1
620   NEXT H
630 NEXT X
640 X=IN:R=20
650 XX=X
660 PLAY"GR64GR64GR60F#F#F#F#R2"
670 FOR H=1 TO 8
680  IF BU(X,H)=1 THEN GOSUB 750:GOTO 700
690  IF BU(X,H)=-1 THEN GOSUB 820
700  GOSUB 880
710 NEXT H
720 R=5 :GOSUB 880
730 FOR N=1 TO 1500:NEXT N:GOSUB 940:END
```

```
740 ' ｽﾌﾟﾗｲﾄ･ﾊﾟﾀｰﾝﾉ ｲﾄﾞｳ
750 FOR G=X*32 TO (X+1)*32
760  PUT SPRITE 0,(G,20*H),3
770  FOR T=1 TO 5:NEXT T
780 NEXT G
790 X=X+1
800 PLAY"T255E#R64E#R64E#R64E#R64GFR64FR64GR64GR64GR64F#F#F#F#"
810 RETURN
820 FOR G=X*32 TO (X-1)*32 STEP -1
830  PUT SPRITE 0,(G,20*H),3
840  FOR T=1 TO 5:NEXT T
850 NEXT G
860 PLAY"T255E#R64E#R64E#R64E#R64GFR64FR64GR64GR64GR64F#F#F#F#"
870 X=X-1:RETURN
880 FOR Q=1 TO R
890  PUT SPRITE 0,(X*32,20*(H)+Q),3
900  FOR V=1 TO 5:NEXT V
910 NEXT Q
920 RETURN
930 ' ｹｯｶﾉ ﾋｮｳｼﾞ
940 PUT SPRITE 0,,0:CLS:COLOR 15,4,7
950 PSET(20,30):PRINT #1,"ﾜｼﾞ の けっか"
960 FOR L=1 TO 5
970  IF L=XX THEN COLOR 9 ELSE COLOR 3
980  Z=XP(L): PSET(20,40+L*20),4:PRINT #1,"(";L;"---";CHR$(&H60+Z);")"
990 NEXT L
1000 PLAY"T255E#R64E#R64E#R64E#R64GFR64FR64GR64GR64GR64F#F#F#F#"
1010 PLAY"T255E#R64E#R64E#R64E#R64GFR64FR64GR64GR64GR64F#F#F#F#"
1020 FOR T=1 TO 1500:NEXT T:RETURN
```

ミュージシャンはキミ！

いまや音楽は、私たちの生活になくてはならないものになっています。外を歩いていても、電車に乗っていても、何かしら音と触れ合う機会は多いはず。でも、実際に自分で音を出すとなると、そういう機会にはなかなかお目にかかれません。

本物のシンセサイザーにはおよびませんが、画面に表示されている鍵盤を見ながら、キーボードをたたく。それだけで3重和音の電子ミュージックを思うように操ることができます。――プログラム作成の合間にちょっとひと息、つかの間のミュージシャン気分を味わってみてはいかが。

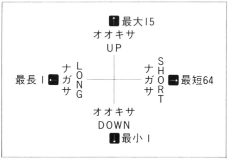


①開始
F5/F10
↓
②音の大きさ
↑
↓
↓
③音の長さ
← →
↓
④ CAPS LOCK
↓
⑤ゲーム
↓
⑥終了
CTRL ＋ STOP

遊び方

●ゲームの進め方

実行すると、画面に鍵盤があらわれる。これが、パソコンミュージックを楽しむために用意されたキーボードだ。これを使って、楽譜を見ながらの演奏、オリジナルな作曲と、自由に音づくりを楽しんでもらいたい。



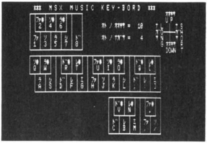
遊び方は簡単。まず、画面の右上に出ている指示に従って、音の大きさと長さを決める。これらの操作は、四つのカーソル移動キーで行い、そ

れぞれ前ページ下図のように対応している。どのカーソル移動キーも、1回押すごとに数値が1ずつ変わるから、演奏する曲に合わせて設定すればよい。

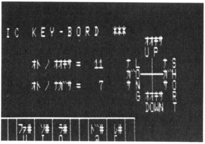
音の調節が終われば、実際の演奏に入る前に CAPS LOCK キーを押して、ロックしておこう。このプログラムは、キーの大文字が画面の鍵盤と対応するようになっているから、そのままだと音が鳴らない。

●演奏

ここまですれば、いよいよスタート。演奏は、画面の鍵盤に書かれた文字や記号のキーを押すだけで、その音が出る。たとえば、Dはドの音。音域は、画面にあるように3オクターブ。同時に三つの音が出せるから、3重和音を使って、本物のシンセサイザーに負けない迫力の演奏が期待できるぞ！



①最初の画面



②音の大きさ、長さを決める

プログラムの解説

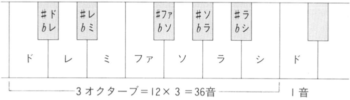
▶画面の初期設定〈行100〜320〉

行100　画面の色を決め、テキスト・モードで1行80桁(けた)に設定し、ファンクション・キーの表示を消す。また、すべての変数を整数型に宣言する。

行110〜320　画面上に、鍵盤と、音の大きさ、長さを決めるためのカーソル移動キーを表示する。

▶音の初期設定〈行330〜450〉

行330〜370　文字型変数S＄に、キーボードの文字を低音から高音の順に取りこ



み、文字列にする。これで、キーボードの文字が、鍵盤のように低音→高音と対応する。

行380〜450 文字の配列変数KY＄(37)を宣言し、その配列に低音→高音の順で取りこむ。音の範囲は、3オクターブとド1音で、計37音。

●音の大きさ、長さの初期設定〈行460〜490〉

行470 音の大きさ、長さ、テンポを、次のように初期値として設定する。

→音の長さ（L）――4分音符♩

→音のテンポ（T）――♩＝120

→音の大きさ（V）――10（最小1〜最大15）

行480 音の大きさの初期値（10）を、画面に表示する。

行490 音の長さの初期値（4）を、画面に表示する。

●音を出す処理〈行500〜750〉

行500 次の、各配列を宣言する。これらは、同時に出せる3音のすべてに対応している。

→キーボードから入力された文字を取りこむ――IN＄

→文字列に対応する音の、マクロの位置を取りこむ――K

→音を出す――P＄

→音の長さ、大きさを変更する――D＄

行510〜530 キーボードから入力された文字を入れるIN＄(1)〜(3)をヌル（何もない状態）にする。

行540〜590 IN＄に入った文字をチェックする。もし何も入力されなかったら(値がヌルのとき)、行510に戻る。同時に出せる3音を、1音ずつチェックする。

行600〜620 IN＄に何も入力されなかった場合、0を代入する。

行630〜650 IN＄に入った文字が、文字列S＄(音のマクロ)の何番めにあるか、**INSTR**（インストリング）関数でそれを探して、値を変数Kの(1)〜(3)に代入する。

行660 K(1)〜K(3)がすべて0であったら、つまり、画面の鍵盤に該当するキーがどれも押されていなければ、行510に戻る。

行670〜700 カーソル移動キーのうち、どれが押されたかを判断する。そして、以下の処理が終わったら行510に戻る。

→K＝38（→キー）――音を短くするとき、行820へ飛ぶ

→K＝39（←キー）――音を長くするとき、行770へ飛ぶ

→K＝40（↑キー）――音を大きくするとき、行920へ飛ぶ

→ K＝41（↓キー）——音を小さくするとき、行870へ飛ぶ

行710〜730　鍵盤に対応する音を、KY＄からP＄に入れる。もし、K(1)〜K(3)が1より小さいか37より大きいとき、つまり鍵盤にないキーが押されたときは、P＄をヌルにしておく。

行740〜750　PLAY（プレイ）命令で音を出し、行510に戻る。

◆音を長くする処理〈行760〜800〉

行770　音の長さNが最小値(1)かどうかをチェックし、最小値のときはNの値を1のままにする。そうでないときは、Nの値を1マイナスする。

行780　Nの値を画面に表示する。

行790　音の長さを変える処理（行970）へ飛ぶ。

◆音を短くする処理〈行810〜850〉

行820　音の長さNが最大値（64）かどうかをチェックし、最大値のときはNの値を64のままにする。そうでないときは、Nの値を1プラスする。

行830　Nの値を画面に表示する。

行840　音の長さを変える処理（行970）へ飛ぶ。

◆音を小さくする処理〈行860〜900〉

行870　音の大きさTが最小値(1)かどうかをチェックし、最小値のときはTの値を1のままにする。そうでないときは、Tの値を1マイナスする。

行880　Tの値を画面に表示する。

行890　音の大きさを変える処理（行970）へ飛ぶ。

◆音を大きくする処理〈行910〜950〉

行920　音の大きさTが最大値(15)かどうかをチェックし、最大値のときはTの値を15のままにする。そうでないときは、Tの値を1プラスする。

行930　Tの値を画面に表示する。

行940　音の大きさを変える処理（行970）へ飛ぶ。

◆音の長さ、大きさを変える処理〈行960〜1060〉

行970〜980　数値型変数を文字型変数に変換する処理を実行する（T→T＄、N→N＄）。

行990〜1010　長さと大きさを変換した文字型変数T＄、N＄をD＄にセットして、音なしでPLAY（プレイ）命令を実行する。こうすることにより、音の長さ、大きさを変更することができる。

行1030〜1050　数値型変数を文字型変数に変換する。

使用変数

D$ 音の大きさ、長さを変更したときに、仮に音を出すときの変数

IN$ キーを押された文字を取りこむ

K キーインされた文字の鍵盤における位置

KY$ 音の高さのマクロ

N 音の長さ。変更できる

P$ PLAY（プレイ）命令で音を出すときの変数。3重和音まで入る

S$ 鍵盤の文字列

T 音のテンポ。変更できる

L 音の長さ

V 音の大きさ



```
99   'MUSIC KEY-BORD
100 COLOR 15,4,4:SCREEN 0:WIDTH 80:KEY OFF:DEFINT A-Z
110 PRINT SPC(17)"***   M S X   M U S I C   K E Y - B O R D   ***"
120 PRINT SPC(15)"┌──┬──┬──┬──┬──┐"SPC(32)"オオキサ"
130 PRINT SPC(15)"│ │ファ#│ ソ#│ ラ#│ │  │"SPC(32)"U P"
140 PRINT SPC(15)"│ │ 2 │ 4 │ 6 │ │  │"SPC(7)"オト ノ オオキサ ="SPC(10)"ナ L  │  ナ S"
150 PRINT SPC(15)"│ └──┴──┴──┘ │  │"SPC(28)"ガ O──┼──ガ H"
160 PRINT SPC(15)"│ ファ│ ソ │ ラ │ シ │ ド │"SPC(7)"オト ノ ナガサ ="SPC(10)"サ N  │  サ O"
170 PRINT SPC(15)"│ 1 │ 3 │ 5 │ 7 │ 8 │"SPC(30)"G オオキサ  R"
180 PRINT SPC(15)"└──┴──┴──┴──┴──┘"SPC(32)"DOWN  T"
190 PRINT SPC(15)"┌─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┐"
200 PRINT SPC(15)"│ ソ#│ ラ#│ │ │ド#│ レ#│ │ │ファ#│ ソ#│ ラ#│ │ │ド#│ レ#│ │"
210 PRINT SPC(15)"│ Q │ W │ │ │ R │ T │ │ │ U │ I │ O │ │ │ @ │ [ │ │"
220 PRINT SPC(15)"└─┬─┴─┬─┘ │ └─┬─┴─┬─┘ │ └─┬─┴─┬─┴─┬─┘ │ └─┬─┴─┬─┘ │"
230 PRINT SPC(17)"│ ラ │ シ │ ド │ レ │ ミ │ ファ│ ソ │ ラ │ シ │ ド │ レ │ ミ │"
240 PRINT SPC(17)"│ A │ S │ D │ F │ G │ H │ J │ K │ L │ ; │ : │ ] │"
250 PRINT SPC(17)"└─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┘"
260 PRINT SPC(45)"┌─┬─┬─┬─┬─┐"
270 PRINT SPC(45)"│ │ド#│ レ#│ │ │ファ#│ │"
280 PRINT SPC(45)"│ │ V │ N │ │ │ . │ │"
290 PRINT SPC(45)"│ └─┴─┘ │ └─┘ │"
300 PRINT SPC(45)"│ ド │ レ │ ミ │ ファ│ ソ │"
310 PRINT SPC(45)"│ C │ B │ M │ , │ / │"
320 PRINT SPC(45)"└─┴─┴─┴─┴─┘"
330 S$=""
340 S$=S$+"CVBNM,./"
350 S$=S$+"QAWSDRFTGHUJIKOL;@:[]"
360 S$=S$+"12345678"
370 S$=S$+CHR$(28)+CHR$(29)+CHR$(30)+CHR$(31)
380 DIM KY$(37)
390 FOR I=1 TO 37
400  READ KY$(I)
410 NEXT I
```

```
420 DATA O3C,O3C+,O3D,O3D+,O3E,O3F,O3F+,O3G,O3G+,O3A,O3A+,O3B
430 DATA O4C,O4C+,O4D,O4D+,O4E,O4F,O4F+,O4G,O4G+,O4A,O4A+,O4B
440 DATA O5C,O5C+,O5D,O5D+,O5E,O5F,O5F+,O5G,O5G+,O5A,O5A+,O5B
450 DATA O6C
460 '   ｵﾄﾉ ｵｵｷｻ ﾅｶﾞｻ ﾄ ｵﾄｦﾀﾞｽ ｼｮﾘ
470 PLAY "L4T120V10","L4T120V10","L4T120V10"
480 T=10:LOCATE 56,3:PRINT USING "###";T
490 N=4:LOCATE 56,5:PRINT USING "###";N
500 DIM IN$(3),K(3),P$(3),D$(3)
510 IN$(1)=""
520 IN$(2)=""
530 IN$(3)=""
540 IN$(1)=INKEY$
550 IF IN$(1) ="" THEN 510
560 IN$(2)=INKEY$
570 IF IN$(1) ="" AND IN$(2)="" THEN 510
580 IN$(3)=INKEY$
590 IF IN$(1)="" AND IN$(2)="" AND IN$(3)="" THEN 510
600 IF IN$(1)="" THEN IN$(1)="0"
610 IF IN$(2)="" THEN IN$(2)="0"
620 IF IN$(3)="" THEN IN$(3)="0"
630 K(1)=INSTR(S$,IN$(1))
640 K(2)=INSTR(S$,IN$(2))
650 K(3)=INSTR(S$,IN$(3))
660 IF K(1)=0 AND K(2)=0 AND K(3)=0 THEN 510
670 IF K(1)=38 OR K(2)=38 OR K(3)=38 THEN GOSUB 820:GOTO 510
680 IF K(1)=39 OR K(2)=39 OR K(3)=39 THEN GOSUB 770:GOTO 510
690 IF K(1)=40 OR K(2)=40 OR K(3)=40 THEN GOSUB 920:GOTO 510
700 IF K(1)=41 OR K(2)=41 OR K(3)=41 THEN GOSUB 870:GOTO 510
710 IF K(1)<1 AND K(1)>37 THEN P$(1)="" ELSE P$(1)=KY$(K(1))
720 IF K(2)<1 AND K(2)>37 THEN P$(2)="" ELSE P$(2)=KY$(K(2))
730 IF K(3)<1 AND K(3)>37 THEN P$(3)="" ELSE P$(3)=KY$(K(3))
740 PLAY P$(1),P$(2),P$(3)
```

```
750 GOTO 510
760 '   ｵﾄｦ ﾅｶﾞｸ ｽﾙ
770 IF N=<1 THEN N=1 ELSE N=N-1
780 LOCATE 56,5:PRINT USING "###";N
790 GOSUB 970
800 RETURN
810 '   ｵﾄｦ ﾐｼﾞｶｸ ｽﾙ
820 IF N=>64 THEN N=64 ELSE N=N+1
830 LOCATE 56,5:PRINT USING "###";N
840 GOSUB 970
850 RETURN
860 '   ｵﾄｦ ﾁｲｻｸ ｽﾙ
870 IF T=<1 THEN T=1 ELSE T=T-1
880 LOCATE 56,3:PRINT USING "###";T
890 GOSUB 970
900 RETURN
910 '   ｵﾄｦ ｵｵｷｸ ｽﾙ
920 IF T=>15 THEN T=15 ELSE T=T+1
930 LOCATE 56,3:PRINT USING "###";T
940 GOSUB 970
950 RETURN
960 '   ｵﾄﾉ ｵｵｷｻ ﾅｶﾞｻ ｦ ｶｴﾙ
970 W=T:GOSUB 1030:T$=W$
980 W=N:GOSUB 1030:N$=W$
990 D$="T120"+"L"+N$+"V"+T$
1000 PLAY D$,D$,D$
1010 RETURN
1020 '   ﾍﾝｽｳ ﾉ ﾍﾝｶﾝ
1030 W$=STR$(W)
1040 L=LEN(W$)-1
1050 W$=RIGHT$(W$,L)
1060 RETURN
```

迷路の館 MSX2対応

ここは、フランスの片田舎にあるギラギラ伯爵の旧館。この館は、一度足を踏み入れると、二度と外には出られない恐怖の館だと人におそれられている。その理由はこうだ。この館には、想像もつかないほどのたくさんの部屋があり、部屋にはいくつもの扉があるという。一つ扉を開いて部屋に入ると、またその部屋にはいくつもの扉がある。それをくり返しているうちに、いつしか人は疲れはて、ついには息たえてしまうということらしい。

①開始
F5/F10
↓
②ゲーム
↑ ← → ↓
↓
③成功／ゲームオーバー
↓
④終了

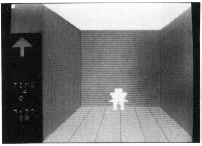
まるで迷路のような、この古ぼけた館——キミははたして、ここからの脱出に挑戦する勇気があるだろうか。運命がキミを見捨てないことを祈る！

遊び方

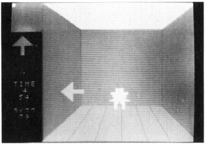
●ゲームの進め方

このゲームは、遊園地によくある鏡の迷路などのコンピュータ版とでもいうべきもの。迷路の中を進み、最も少ない回数で、いちばん短い時間でゴールにたどりついたものが、真の勝利者となる。

実行すると、スタートの部屋があらわれる。キミが進むことのできる方向は、



①最初の画面



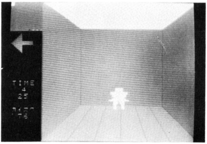
②左に進む

前方と右、左にある扉、それに後方（バック）の4方向。それぞれに対応するカーソル移動キーを使って、館の中を進んでいく。

ただし、実際に通れる扉は、青くぬられたところだけ。灰色の扉はいきどまりで進めないし、三方の扉が灰色の場合は、バックするしかない。

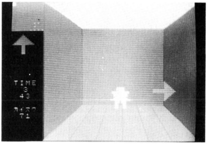
●位置表示パネルをヒントに

画面の左上には、迷路全体の縮小図が表示され、現在地とともに、自分の進んできた部屋がわかるようになっている。現在地は黒点、進んできた部屋は白点で示されるから、これをヒントにゴールへ進んでいこう。

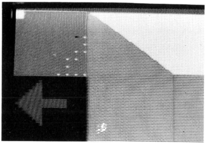


③三方ともいきどまりの部屋

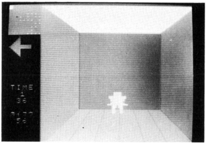
ここで注意するのは、キミの向いている方向が進路によって変わること。方向は、縮小パネルの下の矢印で示されるから、それによって次に進む方向を考える。たとえば、方向の矢印が⇩なら、キミは下を向いていることになる。このとき、画面の上に進みたいときは↓、右なら←のキーを押さなければならない。



④右に進む



⑤現在の位置と方向が左上方に示される



⑥かなり進んだところ



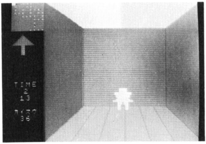
⑦ゲームオーバー

●ゲームオーバー

スタート地点は、迷路全体の右下。ゲームの実行と同時に時計がスタートし、5分経過するとゲームオーバーになってしまう。また、1回進むごとにその回数がカウントされ、80回になるとやはりゲームオーバーになる。

●成功

ゴールにたどりつくと、音楽とともに、「YOU ARE NICE PLAYER（ユー アー ナイス プレイヤー）」と表示される。「キミは、すごいプレイヤーだ」といった意味。また、成功までの経過時間と、進んだ回数が示されるから、最短時間、最少回数に挑戦してもらいたい。



⑧ゴール直前



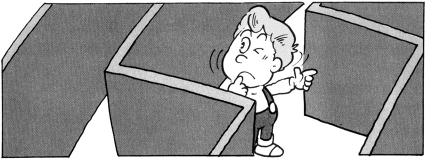
⑨成功

ゲームのコツと注意

◎自分の現在地と向きをいつも確認してから、カーソル移動キーを押すこと。画面だけを見て押すと、思った方向へ進めないことがある。

◎思っていない方向へ進んでしまったときは、■キーでバックすればよい。バックは、三方ふさがりのときだけでなく、いつでもきくのだ。

◎どうしてもゴールできない人は、メモ用紙を用意して、進めるところ、進めないところを記録していくとよい。



ゲームの改造

◘迷路のパターンを変える

迷路のデータ（行970～1070）を変えると、迷路のパターンが変化する。

プログラムの解説

▶初期設定〈行100～450〉

行100～110 使用する画面をビットマップ・グラフィック・モードにし、画面の色を決め、変数や配列を設定し、ファンクション・キーの表示を消す。

行120 グラフィック画面に文字が表示できるようにし、各変数の初期値を設定する。

行130～140 現在地の座標を求める関数を定義する。

行150～160 左、右、前の座標を求める関数を定義する。

行170 進行方向の矢印を表示するための座標を求める関数を定義する。

行180～220 迷路のパターン（データ）を読みこむためのループ。

行230～280 スプライト・パターンのデータを読みこむためのループ。

行290～300 ゲームの残り時間と、トライできる残り回数を表示する。

行310～430 LINE（ライン）文で迷路の部屋をつくり、天井と廊下（ろうか）に色をぬる。

行440 人間のスプライト・パターンを表示する。

行450 位置表示パネルの人間の位置を、スタート地点にセットする。

▶ゲームの実行〈行460～760〉

行460 動いた回数と残り回数をかぞえる。

行470～510 左、右、前の座標を求め、0＝進めるなら、壁を明るい青にぬり、1＝進めないなら、壁を灰色にする。

行520～530 スプライト番号を計算し、そのスプライト・パターンを表示する。

行540～580 カーソル移動キーがどの方向に押されたかを判断する。残り時間、残り回数を計算し、時間オーバーか回数オーバーかを調べ、それぞれの処理へ飛ぶ。

行590 カーソル移動キーが押された方向の数値をチェックする。

行600～620 カーソル移動キーが右、後ろ、左に押されたときは、進む方向を転換する。

行630～650 カーソル移動キーが押された方向に、進めるかどうかを判断する。

進めない方向が入力されたときは、再び方向の入力待ち（行540）に戻る。

行660〜710　進んだ方向の矢印の座標を求め、音を出してそのスプライト・パターンを表示する。

行720〜750　現在地を求め、位置表示パネルに表示する。

行760　ゴールに到達したかどうかを判断する。

●終了画面の処理〈行770〜880〉

行770〜830　ゴールに到達したときの画面処理。音楽を鳴らし、所要時間とトライの回数などを表示する。

行840〜870　ゲーム途中で時間オーバーか回数オーバーになったときの画面処理。音楽を鳴らし、ゲームオーバーになったことを表示する。

行880　プログラムの実行を終了する。

●残り時間、残り回数の表示〈行890〜910〉

●スプライト・パターンと迷路のデータ〈行920〜1070〉

行920　人のスプライト・パターンのデータ。

行930〜960　進む方向を示す矢印のデータ。１行ずつ順に⇧⇨⇩⇦をあらわす。

行970〜1070　迷路のデータ。

使用変数

A　迷路のデータが入る。０なら進むことができる

F　スプライト番号が入る

M　ゲームの経過時間(分)を求めるための変数

M１　ゲームの経過時間(分)

M２　ゲームの残り時間(分)

N　トライの回数

N２　残り回数

P　進む方向（左、右、前、後ろ）が入る

P０　正しい方向に進んだとき、その数値を保存する

P１　進む方向について、入力された数値がそのまま入る

PX、PY　迷路内の座標を、位置表示パネルに対応させて表示するための座標

S　ゲームの経過時間(秒)を求めるための変数

S１　ゲームの経過時間(秒)

S２　ゲームの残り時間(秒)

TX、TY　現在地の座標の右(Ｘ＋１)、左(Ｘ－１)、前（Ｙ＋１）を求めるための変数

X、Y　迷路内の現在地の座標

X０、Y０　正しい方向に進んだとき、その現在地の座標を保存する

```
99 'ﾒｲﾛ･ｹﾞｰﾑ
100 SCREEN 5,3 :COLOR 10,1,2 :CLS
110 DEFINT A-Z :DIM A(12,12) :KEY OFF
120 OPEN "GRP:"FOR OUTPUT AS #1 :TIME=0 :M1=0 :S1=0 :M2=5 :S2=0 :N2=80
130 DEF FNX(P)=VAL(MID$("112221000",P+1,1))-1
140 DEF FNY(P)=VAL(MID$("100122210",P+1,1))-1
150 DEF FNTX(P)=VAL(MID$("012121210101",3*(P+1)/2+I-2,1))-1
160 DEF FNTY(P)=VAL(MID$("101012121210",3*(P+1)/2+I-2,1))-1
170 DEF FNI(P1)=VAL(MID$("1012",(P1+1)/2,1))*30 :DEF FNJ(P1)=VAL(MID$("2101",(P1
+1)/2,1))*25
180 RESTORE 970
190 FOR   I=1 TO 11
200   FOR J=1 TO 11
210   READ A(J,I)
220 NEXT J :NEXT I
230 RESTORE 920
240 FOR   I=0 TO 4 :S$=""
250   FOR J=1 TO 32
260   READ A$:S$=S$+CHR$(VAL("&H"+A$))
270   NEXT J :SPRITE$(I)=S$
280 NEXT I
290 PSET (10,120),15 :PRINT #1,"TIME"
300 PSET (10,160),15 :PRINT #1,"ｶｲｽｳ" :GOSUB 890
310 LINE (0,0)-(48,45),13,BF
320 LINE (0,0)-(10,5),15,BF
330 LINE (48,0)-(255,211),,B
340 LINE (98,43)-(204,157),,B
350 LINE (48,0)-(98,43)
360 LINE (204,43)-(255,0)
370 LINE (48,211)-(98,157)
380 LINE (255,211)-(204,157):PAINT (135,180),11,10
390 LINE (81,211)-(114,157)
```

```
400 LINE (116,211)-(132,157)
410 LINE (151,211)-(151,157)
420 LINE (185,211)-(169,157)
430 LINE (216,211)-(187,157) :PAINT(130,30),15,10
440 PUT SPRITE 5,(132,130),15,0
450 X=10 :Y=10 :P=1 :PSET (47,44),1 :PX=47 :PY=44 :M2=4
460 P0=P :N=N+1 :N2=80-N
470 FOR I=0 TO 2
480   TX=X+FNTX(P):TY=Y+FNTY(P)
490   IF A(TX,TY)=0 THEN C=5  ELSE C=14
500   PAINT(70*(I+1),100),C,10
510 NEXT I
520 X0=X :Y0=Y :F=(P+1)/2
530 PUT SPRITE 8,(5,47),2,F
540 P=STICK(0) :P1=P :T=TIME :M=INT(T/60^2) :S=INT(T/60)
550 IF  M>M1  THEN M1=M1+1 :M2=4-M1
560 S1=S-M1*60 :S2=60-S1 :GOSUB 890
570 IF M1>=5 THEN 840
580 IF N>79  THEN 840
590 IF P=1 OR P=3 OR P=5 OR P=7 THEN 600 ELSE 540
600 IF P0=3 THEN P=P+2:IF P>7 THEN P=1
610 IF P0=5 THEN P=P+4:IF P>7 THEN P=P-8
620 IF P0=7 THEN P=P-2:IF P<0 THEN P=7
630 X=X+FNX(P) :Y=Y+FNY(P)
640 IF X>0 AND X<12 AND Y>0 AND Y<12  THEN 650 ELSE 540
650 IF A(X,Y)<>0 THEN X=X0:Y=Y0:GOTO 540
660 I=30-FNI(P1) :J=25-FNJ(P1) :K=(P1+1)/2
670 FOR L=1 TO 2
680   PUT SPRITE 1,(131+L*I,120+L*J),7,K
690   PLAY "O5A4"
700   IF PLAY(0) THEN 700
710 NEXT L
720 PUT SPRITE 1,(255,209)
```

```
730 PRESET (PX,PY),15
740 PX=(X-1)*5+2 :PY=(Y-1)*5-1
750 PSET (PX,PY),1
760 IF X=2  AND  Y=2  THEN 770 ELSE 460
770 SCREEN 1,0:COLOR 15,4,11
780 PLAY "O4G4O5C8.R32C8R32C2 O4G4O5D4O4B8O5C2"
790 LOCATE 2,8 :PRINT "*** YOU ARE NICE PLAYER ***"
800 LOCATE 10,12 :PRINT "YOUR SCORE "
810 LOCATE 10,15 :PRINT M1 :LOCATE 12,15 :PRINT "分"
820 LOCATE 14,15 :PRINT S1 :LOCATE 17,15 :PRINT "秒"
830 LOCATE 10,18 :PRINT N  :LOCATE 14,18 :PRINT "kai" :GOTO 880
840 SCREEN 1,0:COLOR 15,4,11
850 PLAY "O4C2R16C4R16C8R16C2RE-4D4R16D4C4R18C4O3B4O4C1"
860 LOCATE 10,10 :PRINT "GAME OVER "
870 LOCATE 8,14 :PRINT "さ゛んねんて゛した またと゛うそ゛"
880 END
890 PSET (15,130),15:PRINT #1,M2
900 PSET (10,140),15:PRINT #1,S2
910 PSET (10,170),15:PRINT #1,N2 :RETURN
920 DATA 03,03,03,1F,1F,0F,0F,07,0F,3F,07,07,06,06,06,06,E0,E0,E0,FC,FC,F8,F8,F0
,F8,FE,70,70,30,30,30,30
930 DATA 00,00,01,03,07,0F,1F,3F,7F,01,01,01,01,01,01,01,00,80,C0,E0,F0,F8,FC,FE
,FF,C0,C0,C0,C0,C0,C0,C0
940 DATA 01,01,01,01,01,01,FF,FF,FF,01,01,01,01,01,01,00,00,80,C0,E0,F0,F8,FC,FE
,FC,F8,F0,E0,C0,80,00,00
950 DATA 03,03,03,03,03,03,03,FF,7F,3F,1F,0F,07,03,01,00,80,80,80,80,80,80,80,FE
,FC,F8,F0,E0,C0,80,00,00
960 DATA 00,01,03,07,0F,1F,3F,7F,3F,1F,0F,07,03,01,00,00,80,80,80,80,80,80,FF,FF
,FF,80,80,80,80,80,80,00
970 DATA 1,1,1,1,1,1,1,1,1,1,1
980 DATA 1,0,0,1,0,0,0,0,0,0,1
990 DATA 1,1,0,0,1,0,0,1,0,0,1
1000 DATA 1,0,0,1,0,1,0,0,0,1,1
```

```
1010 DATA 1,1,0,0,0,1,0,1,0,0,1
1020 DATA 1,0,0,1,0,0,0,0,1,0,1
1030 DATA 1,0,1,0,1,0,1,0,0,0,1
1040 DATA 1,0,0,0,1,0,1,0,0,1,1
1050 DATA 1,1,0,1,0,0,1,0,1,1,1
1060 DATA 1,0,0,0,0,1,0,0,0,0,1
1070 DATA 1,1,1,1,1,1,1,1,1,1,1
```

ポーカーゲーム

ギャンブル性が高く、トランプの中でも人気の高いポーカー。役づくりとともに、いわゆるポーカーフェイスでいかに相手をだますかというテクニックも、ポーカーのおもしろさになくてはならない要素です。

このゲームは、パソコン相手に遊ぶ一人ポーカー。もちろん、何人かが順番に競い合うこともできます。キミのポーカーフェイスがコンピュータにどこまで通じるか、試してみるのもよいでしょう。

①開始
F5/F10
↓
②ゲーム
Y/N チェンジの選択
↓
CAPS LOCK
↓
C RETURN
1枚ごとのチェンジ
↓
③終了
役の判定のあと、自動的に終わる



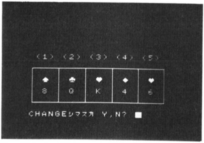
遊び方

◉ゲームの進め方

実行すると、最初に配られる５枚のカードが、画面に表示される。同時に、チェンジするかどうかを聞いてくるので、そのままでよければ N RETURN 、１枚でもチェンジするなら Y RETURN と押す。

チェンジの場合は、ここからがカンの働かせどころ。いま、カーソルは？マークとともに左のカードの上にある。そのカードを交換するなら C RETURN と押す。このとき、Cは大文字しかきかないから、ゲームに入る前に CAPS LOCK キーをロックしておこう。

そのカードをチェンジしたくない場合は、 RETURN キーだけを押す。



①最初の画面

こうして、チェンジまたはそのままの

どちらかを決めると、カーソルが一つずつ右にずれていく。5枚のカードそれぞれで選択し、5枚めがチェンジかどうかを決めたところで、選択終了となる。

最後の RETURN キーインで、Cが表示されたカードがチェンジされ、ゲームの結果が示される。

●役の表示

どの役ができたかによって、次のように表示される。

➜役なし→ブタダヨ——

➜ワンペア→ONE PAIR　ダヨーン

➜ツーペア→TWO PAIR　ジャネーカ

➜スリーカード→THREE CARD　ダゼイ!!

➜ストレート→ストレート　ストレート　ストレート

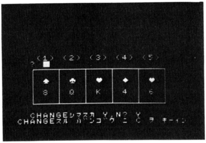
➜フラッシュ→フラッシュ

➜フルハウス→FULL HOUSE　シンジランナーイ

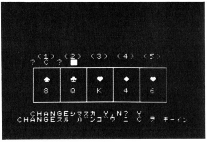
➜ストレートフラッシュ→＊＊＊ストレートフラッシュ＊＊＊

➜フォーカード→FOUR CARD　ダワーイ

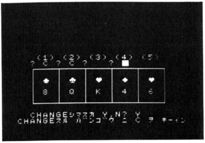
➜ロイヤルストレートフラッシュ→ミテハイケナイモノヲ　アナタハ　ミタ



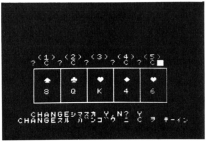
②チェンジの画面



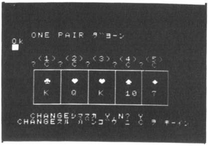
③チェンジ



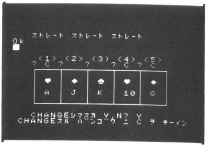
④チェンジしない



⑤選択終了



⑥結果



⑦ストレートの結果表示

ゲームのコツと注意

◎チェンジの選択で、一度移動したカーソルは、バックできない。時間制限はないから、慎重に考えてから選ぼう。

◎何人かで遊ぶなら、こんなゲームはどうだろう。一人10回ずつゲームして、それを記録しておく。ほかの人は、その結果を覚えておく。全員が10回のゲームを終わったら、各自はその10回の結果をバラバラにして、任意に1～10の番号をつける。そして、番号順にその結果（役）を見せ合う。最も高い役の人が勝ちで、10回戦のトータルを競う。



プログラムの解説

▶初期設定とカードわくの表示〈行100～200〉

行100　使用する画面をテキスト・モードにして、表示桁(けた)数を決め、ファンクション・キーの表示を消す。

行110　乱数の発生が、実行のたびに変わるようにする。

行120　各配列変数を宣言する。

行130～200　カードのわくを表示し、左から(1)～(5)の順に番号をつける。

▶カードの表示〈行210～260〉

行220　カードを表示するためのループ。

行230　カードを配るサブルーチンを呼び出して実行し、配られたカードの数字とマークを変数に保存しておく。

行240　カードのマーク（♠♥♣♦）を表示する。

行250　カードの数字（A～K）を表示する。

行260　以上を5回くり返して、ループを終わる。

●カードをチェンジするかどうかの選択〈行270～300〉

行280　「CHANGE　シマスカ　Y，N」を表示し、文字をキーインさせる。

行290　キーインされた文字が大文字のNまたは小文字のnなら、役の判定処理（行700）へ飛ぶ。

行300　キーインされた文字が「Y」か「y」でないときは行280に戻り、キーインを待つ。空白を座標（3，19）に表示しているのは、まちがってキーインされた文字を消すための処理。

●チェンジするカードの選択〈行310～360〉

行320　説明文を表示する。

行330～360　それぞれのカードをチェンジするかどうかをキーインしていくためのループ。「C」か RETURN キー以外の文字がキーインされたときは、その文字を消し、もう一度キーインを待つ。

●カードのチェンジ〈行370～460〉

行380～400　チェンジするカードを、配列Sに保存するための処理。

行410　カードをチェンジするためのループの開始。

行420　カードをチェンジしないとき、行460へ飛ぶ。

行430　「C」がキーインされたとき、新しいカードに変える処理（行530）へ飛ぶ。

行440～450　カードの表示。

行460　以上を5回くり返し、ループを終わる。

●役の判定〈行470～510〉

行470～490　役を判定する準備として、最初に配られたカードのうち、チェンジするカードのところの配列を0にしておく。

行500　行700へ飛んで、役の判定を行う。

行510　プログラムの実行を終了する。

●カードを配るサブルーチン〈行520～680〉

行530　1～13の乱数を発生させる。

行540　発生した乱数が1、11、12、13のとき、それぞれをA、J、Q、Kに変える処理（行640）へ飛ぶ。

行550〜590　1〜4の乱数を発生させ、その数字に♠♥♣♦を割りあてる。

行600　すでに配られたカードかどうかをチェックし、そのときは行530に戻って、もう一度カードを出す。

行610　役を判定するための配列Aの、配られたカードの場所に1をセットする。

行640〜680　1をA、11をJ、12をQ、13をKに変えるサブルーチン。

➡役の判定処理〈行690〜1100〉

役の判定には、配列A（13, 4）を使う。最終的に決定したカードのところには1がセットされていて、たとえば5枚のカードが♥A、♣A、♠4、♦J、♣Qのときは、図のようになる。

	A	2	3	4	5	6	7	8	9	10	J	Q	K
♠				1									
♥	1												
♣	1											1	
♦											1		

ここで、配列の縦方向と横方向で1の数字を加算していけば、次のように役を判定できることがわかる。

①まず、縦方向の加算値で判定する。

➡加算値2が一組→ワンペア（図の例）

➡加算値3が一組→スリーカード

➡加算値4が一組→フォーカード

➡加算値2が二組→ツーペア

➡加算値が2と3→フルハウス

②次に、横方向の加算値で判定する。

➡加算値が5→フラッシュ

③ストレートは、縦方向の加算値1が5個で、それが連続しているとき。

以下、具体的には次のとおり。

行700〜720　縦方向の加算を13回くり返すループ。

行730〜770　縦方向の加算結果を調べるためのループ。行740でワンペア、行750でスリーカード、行760でフォーカードが、それぞれできているかどうかを判定する。

行780　フルハウスの判定と表示。

行790〜820　役の表示。

行830　以上の役ができていたら、行1100へ飛ぶ。

行840〜880　ストレートを判定するために、配列変数 GN を設定し、そこに配られたカードをセットするループ。GN には、マークに関係なく、配られたカードの数の場所に 1 がセットされる。

行890　数が連続しているかどうかを調べるために、変数 YW を初期化する。

行900〜930　カードの数が連続しているかどうかの判定。連続していないと 0 が入るので、YW が 5 ならストレートということになる。

行940　ロイヤルストレートの可能性を判定。可能性がなければ行970へ飛ぶ。

行950　1（A）が配られたかどうかの判断。配られているなら、YW に 1 をプラスする。

行960　ロイヤルストレートのときの処理。

行970〜990　横方向の加算のため、配列変数 GK を 0 にしておく。

行1000〜1040　横方向の加算を 4 回くり返すループ。

行1050　フラッシュができたかどうかの判断。

行1060　ロイヤルストレートフラッシュの判定と表示。

行1070　ストレートフラッシュの判定と表示。

行1080　ストレートの判定と表示。

行1090　フラッシュ、ブタ（役なし）の判定と表示。

使用変数

A　52枚のカードのうち、どれが選ばれたかを調べるための 2 次元配列

CR　最初に配られたカードを入れておく配列

FL　フラッシュのとき、1 をセットする

GN、GK　役を判定するために設定する配列

G$　1、11、12、13のとき、A、J、Q、Kに変えるための文字型変数

IX、IS　IXはカードの数字、ISはマークで、A（IX，IS）となり、配列Aの場所をあらわす

P2　ワンペアのとき、1をセットする

P3　スリーカードのとき、1 をセットする

P4　フォーカードのとき、1 をセットする

P$　カードをチェンジするかどうかを入れる文字型変数

RO　ロイヤルストレートのとき、1 をセットする

S　最初に配られたカードのマークと数字を保存する

ST　ストレートのとき、1 をセットする

W$　カードのマークを入れる文字型変数

```
99 '      "       POKER   GAME       "
100 SCREEN 1:WIDTH 28:KEY OFF
110 Z=RND(-TIME)
120 DIM A(13,4):DIM GN(13):DIM GK(4):DIM CR(5,2):DIM P$(5)
130 LOCATE 3,11:PRINT  "┌───┬───┬───┬───┬───┐"
140 LOCATE 3,12:PRINT  "│   │   │   │   │   │"
150 LOCATE 3,13:PRINT  "│   │   │   │   │   │"
160 LOCATE 3,14:PRINT  "│   │   │   │   │   │"
170 LOCATE 3,15:PRINT  "│   │   │   │   │   │"
180 LOCATE 3,16:PRINT  "│   │   │   │   │   │"
190 LOCATE 3,17:PRINT  "└───┴───┴───┴───┴───┘"
200 LOCATE 4,9:PRINT  "(1) (2) (3) (4) (5)"
210 '  ｶｰﾄﾞﾉ ﾋｮｳｼﾞ
220 FOR Q=1 TO 5
230   GOSUB 530:CR(Q,1)=IX:CR(Q,2)=IS
240   LOCATE 4*Q+1,13:PRINT W$
250   LOCATE 4*Q,15:PRINT LEFT$(IX$,3)
260 NEXT Q
270 '  ｹﾞｰﾑﾉ ｶｲｼ
280 LOCATE 3,19:INPUT "CHANGEｼﾏｽｶ Y,N";ZZ$
290 IF ZZ$="N" OR ZZ$="n" THEN 500
300 IF ZZ$<>"Y" AND ZZ$<>"y" THEN LOCATE 3,19:PRINT SPACE$(LEN(ZZ$)+16):GOTO 280
310 '  ｶｰﾄﾞﾉ ｾﾝﾀｸ
320 LOCATE 1,20:PRINT  "CHANGEｽﾙ ﾊﾞﾝｺﾞｳ ﾆ C ｦ ｷｰｲﾝ"
330 FOR P=1 TO 5
340   LOCATE P*4-1,10:INPUT P$(P)
350   IF P$(P)<>"C" AND P$(P)<>"" THEN LOCATE P*4+1,10:PRINT SPACE$(LEN(P$(P))):
P$(P)="":GOTO 340
360 NEXT P
370 '  ｶｰﾄﾞﾉ ﾍﾝｺｳ
380 FOR P=1 TO 5
390   IF P$(P)="C" THEN S(P,1)=CR(P,1):S(P,2)=CR(P,2)
400 NEXT P
```

```
410 FOR P=1 TO 5
420   IF P$(P)="" THEN GOTO 460
430   IF P$(P)="C" THEN GOSUB 530
440   LOCATE P*4+1,13:PRINT W$
450   LOCATE P*4,15:PRINT IX$
460 NEXT P
470 FOR P=1 TO 5
480   A(S(P,1),S(P,2))=0
490 NEXT P
500 GOSUB 700
510 END
520 '  ｶｰﾄﾞｦ ｸﾊﾞﾙ
530 IX=INT(RND(1)*13+1)
540 IF IX=1 OR IX=11 OR IX=12 OR IX=13 THEN GOSUB 640 ELSE IX$=STR$(IX)+" ":IX$=
LEFT$(IX$,3)
550 IS=INT(RND(1)*4+1)
560 IF IS=1 THEN W$="♠"
570 IF IS=2 THEN W$="♥"
580 IF IS=3 THEN W$="♣"
590 IF IS=4 THEN W$="♦"
600 IF A(IX,IS)=1 THEN GOTO 530
610 A(IX,IS)=1
620 RETURN
630 '
640 IF IX=1 THEN IX$=" A "
650 IF IX=11 THEN IX$=" J "
660 IF IX=12 THEN IX$=" Q "
670 IF IX=13 THEN IX$=" K "
680 RETURN
690 '  ﾔｸﾉ ﾊﾝﾃｲ
700 FOR F=1 TO 13
710   GN(F)=A(F,1)+A(F,2)+A(F,3)+A(F,4)
720 NEXT F
```

```
730 FOR F=1 TO 13
740   IF GN(F)=2 THEN P2=P2+1
750   IF GN(F)=3 THEN P3=1
760   IF GN(F)=4 THEN P4=1
770 NEXT F
780 IF P2=1 AND  P3=1  THEN LOCATE 3,5:PRINT "FULL HOUSE ｼﾝｼﾞﾗﾝﾅｰｲ":GOTO 830
790 IF P2=1 THEN LOCATE 3,5:PRINT "ONE PAIR ﾀﾞﾖｰﾝ"
800 IF P2=2 THEN LOCATE 3,5:PRINT "TWO PAIR ｼﾞｬﾈｰｶ"
810 IF P3=1 THEN LOCATE 3,5:PRINT "THREE CARD ﾀﾞｾﾞｲ!!"
820 IF P4=1 THEN LOCATE 3,5:PRINT "FOUR CARD ﾀﾞ ﾜｰｲ"
830 IF P2=1 OR   P2=2   OR  P3=1   OR   P4=1 THEN  GOTO 1100
840 FOR F1=1 TO 4
850   FOR F2=1 TO 13
860     IF A(F2,F1)=1 THEN GN(F2)=1
870   NEXT F2
880 NEXT F1
890 YW=0
900 FOR F1=1 TO 13
910   IF GN(F1)=1 THEN YW=YW+GN(F1) ELSE YW=0
920   IF YW=5 THEN F1=13:ST=1
930 NEXT F1
940 IF GN(13)<>1 OR YW<>4 THEN 970
950 IF GN(1)=1 THEN YW=YW+1
960 IF YW=5 THEN RO=1:ST=1
970 FOR F1=1 TO 4
980   GK(F1)=0
990 NEXT F1
1000 FOR F1=1 TO 4
1010   FOR F2=1 TO 13
1020     GK(F1)=GK(F1)+A(F2,F1)
1030   NEXT F2
1040 NEXT   F1
```

```
1050 IF GK(1)=5 OR GK(2)=5 OR GK(3)=5 OR GK(4)=5 THEN FL=1
1060 IF RO=1 AND ST=1 AND FL=1 THEN LOCATE 3,5:PRINT "ﾐﾃﾊｲｹﾅｲﾓﾉｦ ｱﾅﾀﾊ ﾐﾀ":GOTO 1
100
1070 IF ST=1 AND FL=1 THEN LOCATE 3,5:PRINT "******ｽﾄﾚｰﾄﾌﾗｼｭ******":GOTO 1100
1080 IF ST=1 THEN LOCATE 3,5:PRINT "ｽﾄﾚｰﾄ ｽﾄﾚｰﾄ ｽﾄﾚｰﾄ":GOTO 1100
1090 IF FL=1 THEN LOCATE 3,5:PRINT "ﾌﾗｯｼｭ" ELSE LOCATE 3,5:PRINT "ﾌﾞﾀﾀﾞﾖ -------
-"
1100    RETURN
```

逃亡ゲーム

逃げては追いつかれ、また逃げては追いつかれ――まるで相手からはこちらの動きが見えるように、どこへ逃げてもインベーダの魔の手が伸びてきます。

インベーダから逃れる手段はただひとつ、バリケードをつくることだけです。ただし、インベーダはバリケードも食い荒らしてしまうので、その間少し時間をかせげるだけ。キミのはてしない逃亡の旅が続きます。



①開始

F5/F10

↓

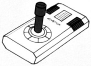
②パワーキーイン

1 ～ 100＋RETURN

↓

③ゲーム

ジョイスティックとトリガボタン



↓

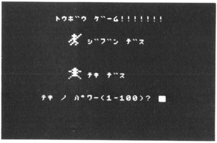
④終了

遊び方

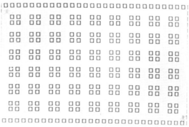
●敵のパワー決定

実行するとゲームの説明画面になり、敵のパワーをいくつにするかを聞いてくる。パワーは１～100の範囲で、数字が大きくなるほど敵の力が強くなる。

最初に遊ぶ場合は、10ぐらいが適当だろう。数字をキーインしたら、RETURN



①最初の画面。敵のパワーをキーイン



②ゲーム画面

キーを押す。これでゲーム開始。

●ゲームの進め方

スタートしたら、画面に迷路のような道ができ、左上にインベーダ、右下に人間があらわれる。これが、キミだ。インベーダはすぐに追いかけてくるから、ジョイスティックを上下左右に操って、うまく逃げよう。

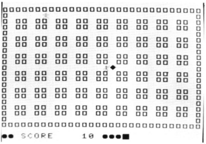
インベーダにつかまりそうになったら、トリガボタンを押す。◆のバリケードがあらわれ、画面下方に得点が表示される。

●インベーダの動き

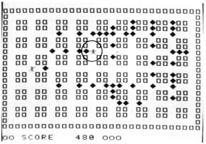
インベーダは、逃げるキミをめざして、最短コースを進んでくる。そして、そこにバリケードがあると、食うか食わないかを判断する。すぐに食ってしまうこともあるが、だいたいは少し迷って色が暗くなる。その間に、安全なところに逃げておこう。

●ゲームの終了とスコア

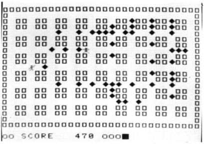
逃げそこなって、インベーダにつかまってしまったら、そこでゲームオーバー。残ったバリケードが、1個につき10点としてスコアに換算される。



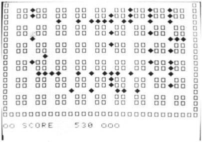
③バリケードを落とす



④バリケードを食うかどうかの判断



⑤バリケードを食う

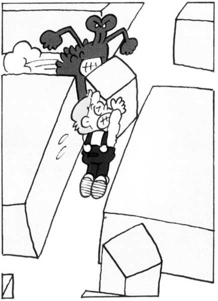


⑥残念、つかまった



⑦スコアと終了画面

ゲームのコツと注意



◎インベーダにつかまらないようにするには、バリケードを多く落とすこと。しかも、その数が得点に結びつく。ただし、インベーダはバリケードを食って先に進めるけど、人間は通れない。へたにつくると、自分の逃げ道をふさいでしまうことになる。

◎バリケードは、できるだけ画面の中央のほうからつくるのがコツ。道の周囲は、逃げ道として、できるだけあけておこう。

◎へんなところにバリケードをつくってしまったら、インベーダをそこにおびき寄せ、食わしてすぐに逃げる、という高等テクニックもある。

プログラムの解説

▶ゲーム処理の流れ 〈行100〜130〉

行100　初期画面の表示を呼び出して実行。

行110　人間を動かす処理を呼び出して実行。

行120　乱数を発生させ、インベーダのパワー以上の数字が出たとき、インベーダを動かす処理を呼び出して実行。

行130　行110に戻って、処理をくり返す。

▶初期画面の表示 〈行140〜240〉

行150　使用する画面をテキスト・モードにし、表示桁(けた)数、色を決め、ファンクション・キーの表示を消す。

行160　乱数の発生が、実行のたびに変わるようにする。

行170　タイトルを表示する。

行180　スプライト・パターン作成の処理を呼び出して、実行する。

行190〜200　人間のスプライト・パターンと、「ジブン　デス」という文字を表示。

行210〜220　インベーダのスプライト・パターンと、「テキ　デス」という文字を表示。

行230〜240　インベーダのパワーをキーインさせるための処理。1〜100の範囲内とする。

▶壁と道の表示〈行250〜420〉

行260　ゲーム画面の設定。

行270　すべての変数を整数型に宣言する。

行280　変数ZZに初期値（0.7）をセットする。

行290　配列変数を設定する。

行300〜410　人間とインベーダが通る道をつくるループ。

行420　人間とインベーダのスプライト・パターンのデータを読みこみ、パターンを作成する。

▶割りこみ先の設定〈行430〜480〉

行440　スプライト・パターンが衝突したとき、行980（終了処理）へ飛ぶ。

行460　ジョイスティックのトリガボタンが押されたとき、行920へ飛ぶ。

行470　スプライト・パターンが衝突したときの割りこみを可能にする。

行480　トリガボタンが押されたときの割りこみを可能にする。

▶人間とインベーダの表示とジョイスティックの制御〈行490〜570〉

行500〜520　人間とインベーダのスプライト・パターンを表示する。

行550〜560　ジョイスティックが上下､左右の4方向にしかきかないようにする。

▶人間の制御〈行580〜700〉

行590　ジョイスティックが押された方向を調べる。

行600　どの方向にも押されていなければ、この処理を実行しない(人間は動かない)。

行610　人間が進む方向のセット。TX、TYは仮のX座標、Y座標で、XV、YVには行550〜560によって、下表のような値が与えられている。これが座標の増しぶんとなり、たとえばジョイスティックの押された方向（SV）が右（＝3）なら、X座標が1プラスされ、人間が右に進むことになる。

SVの値	1	2	3	4	5	6	7	8
XV	0	0	1	0	0	0	−1	0
YV	−1	0	0	0	1	0	0	0

行620～650　人間が壁に衝突したときの処理。座標の値を修正する。

行660～670　進む方向が道ではないとき、人間を動かさない。

行680～690　人間を移動させた位置で表示する。

⊃インベーダの制御〈行710～900〉

行720～730　人間とインベーダの相対位置の計算。

行740　人間とインベーダが横一線にも縦一線にも並んでいないときは行790へ、横一線のときは行770へ、どちらでもないとき（縦一線）は次の行へ飛ぶ。

行750～860　インベーダが進める方向と人間との相対的な位置関係を調べ、人間の方向（TX、TY）に数値をセットする。

行870　インベーダが進むところにバリケードがあれば、0以上1未満の乱数を発生させ、それがZZの値(0.7)より小さければ、インベーダの色を暗い赤にする。乱数の値が0.7以上なら、バリケードを消し、スコアを10マイナスし、新しくスコアを表示する。

行880～890　インベーダを移動させた位置で表示する。

⊃バリケードの表示〈行910～960〉

行920　すでにバリケードが置かれているかどうかをチェックし、置かれていれば表示処理をしない。

行930　バリケードとして◆を表示し、音を出す。

行940　バリケードがあるかどうかをチェックするための配列ADに、1をセットしておく。

行950　スコアを10プラスし、表示する。

行960　行110に戻る。

⊃ゲームオーバーの処理〈行970～1040〉

行980～990　スプライトの衝突による割りこみを禁止し、終了画面を設定して音を出す。

行1000～1040　「GAME OVER（ゲームオーバー）」とスコアを表示し、プログラムの実行を終わる。

⊃人間とインベーダの作成〈行1050～1150〉

行1060～1120　データを読みこみ、人間とインベーダのスプライト・パターンをつくるループ。行1090で行1140のデータから最初の8個を読みこみ、パターン0にしまう（インベーダ）。同様にしてあとの8個を読みこみ、人間（パターン1）をつくる。

行1140～1150　スプライト・パターンのデータ。

使用変数

- **AD**　そこが道かどうかということや、バリケードがあるかないかの情報を入れる配列変数
- **CX、CY**　インベーダの座標
- **SC**　スコア
- **SX、SY**　人間の座標
- **S$**　スプライト・パターンのデータ用の文字列
- **TX、TY**　スプライト・パターンが移動した座標
- **WB**　インベーダと人間の座標の差が、正か負かを入れる
- **WX、WY**　インベーダと人間の座標の差
- **XV、YV**　ジョイスティックの初期値
- **Z**　インベーダのパワー
- **ZZ**　インベーダがバリケードを食う確率を決める

●ジョイスティックがないときは

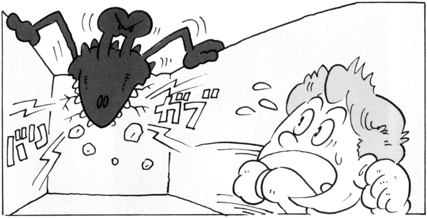
次のようにプログラムを修正すると、ジョイスティックの代わりにカーソル移動キー、トリガボタンの代わりにスペース・キーが使えるようになる。

①行460の「GOSUB」に続くコンマ（,）をとる。

②行480、590、980の「STRIG」または「STICK」に続くカッコ内の1を、0に変える。

➡人間の移動→←→↑↓

➡バリケードをつくる→



```
99 'ﾄｳﾎﾞｳ ｹﾞｰﾑ
100 GOSUB 150
110 GOSUB 590
120 IF RND(1)>Z/100 THEN GOSUB 720
130 GOTO 110
140 '-------- ﾒﾆｭｰ ｶﾞﾒﾝ ﾋｮｳｼﾞ -----------
150 SCREEN 1,1:WIDTH 32:COLOR 15,1,1:CLS:KEY OFF
160 DD=RND(-TIME)
170 LOCATE 8,1:PRINT "ﾄｳﾎﾞｳ ｹﾞｰﾑ!!!!!!!"
180 GOSUB 1060
190 PUT SPRITE 1,(10*8,4*8),5,1
200 LOCATE 13,5:PRINT "ｼﾞﾌﾞﾝ ﾃﾞｽ"
210 PUT SPRITE 0,(10*8,10*8),8,0
220 LOCATE 13,11:PRINT "ﾃｷ ﾃﾞｽ"
230 LOCATE 6,15:INPUT "ﾃｷ ﾉ ﾊﾟﾜｰ(1-100)";Z
240 IF Z > 100 OR Z=<0 THEN GOTO 230 ELSE Z=100-Z
250 '------- ｶﾍﾞ ﾄ ﾐﾁ ﾉ ﾋｮｳｼﾞ ---------
260 SCREEN 1,0:WIDTH 32:COLOR 1,10,10
270 DEFINT A-Y
280 ZZ=.7
290 DIM AD(30,21)
300 FOR I=1 TO 21
310  FOR J=1 TO 30
320   LOCATE J,I:PRINT "ロ":AD(J,I)=9
330  NEXT J
340 NEXT I
350 FOR I=2 TO 20
360  M=I MOD 3
370  IF M=2 THEN M=1 ELSE M=3
380  FOR J=2 TO 29 STEP M
390   LOCATE J,I:PRINT " ":AD(J,I)=0
400  NEXT J
410 NEXT I
```

```
420 RESTORE 1140:GOSUB 1060
430 '------ ｼｮｳﾄﾂ ﾉ ﾜﾘｺﾐｻｷ ------------
440 ON SPRITE GOSUB 980
450 '------ ﾄﾘｶﾞ ﾎﾞﾀﾝ ﾉ ﾜﾘｺﾐｻｷ --------
460 ON STRIG GOSUB ,920
470 SPRITE ON
480 STRIG(1) ON
490 '------ ﾆﾝｹﾞﾝ ﾄ ｲﾝﾍﾞｰﾀﾞ ﾉ ﾋｮｳｼﾞ ---
500 CX=2:CY=2:SX=29:SY=20
510 PUT SPRITE 0,(CX*8,CY*8),8,0
520 PUT SPRITE 1,(SX*8,SY*8),5,1
530 '------ ｼﾞｮｲ ｽﾃｨｯｸ ﾉ ｼｮｷｾｯﾃｲ    ----
540 DIM XV(8),YV(8)
550 XV(1)=0:XV(2)=0:XV(3)=1:XV(4)=0:XV(5)=0:XV(6)=0:XV(7)=-1:XV(8)=0
560 YV(1)=-1:YV(2)=0:YV(3)=0:YV(4)=0:YV(5)=1:YV(6)=0:YV(7)=0:YV(8)=0
570 RETURN
580 '------- ﾆﾝｹﾞﾝ ﾉ ｾｲｷﾞｮ ------------
590 SV=STICK(1)
600 IF SV=0 THEN RETURN
610 TX=SX+XV(SV):TY=SY+YV(SV)
620 IF TX<2 THEN TX=2
630 IF TX>29 THEN TX=29
640 IF TY<2 THEN TY=2
650 IF TY>20 THEN TY=20
660 IF AD(TX,TY)=1 THEN RETURN
670 IF AD(TX,TY)=9 THEN RETURN
680 SX=TX:SY=TY
690 PUT SPRITE 1,(SX*8,SY*8),5,1
700 RETURN
710 '------- ｲﾝﾍﾞｰﾀﾞ ﾉ ｾｲｷﾞｮ ----------
720 TX=SX-CX:TY=SY-CY
730 WX=ABS(TX):WY=ABS(TY):WB=SGN(WX-WY)
740 IF WX>0 AND WY>0 THEN 790 ELSE IF WX>0 AND WY=0 THEN 770
```

```
750 IF AD(CX,CY-1)=9 AND AD(CX,CY+1)=9 THEN IF TY>0 THEN IF AD(CX-1,CY+1)=9 THEN
TX=1:TY=0 ELSE TX=-1:TY=0 ELSE IF AD(CX-1,CY-1)=9 THEN TX=1:TY=0 ELSE TX=-1:TY=0
 ELSE IF TY>0 THEN TX=0:TY=1 ELSE TX=0:TY=-1
760 GOTO 860
770 IF AD(CX-1,CY)=9 AND AD(CX+1,CY)=9 THEN IF TX>0 THEN IF AD(CX+1,CY+1)=9 THEN
 TX=0:TY=-1 ELSE TX=0:TY=1 ELSE IF AD(CX-1,CY+1)=9 THEN TX=0:TY=-1 ELSE TX=0:TY=
1 ELSE IF TX>0 THEN TX=1:TY=0 ELSE TX=-1:TY=0
780 GOTO 860
790 IF TX>0 AND TY>0 THEN IF WB>0 THEN IF AD(CX+1,CY)=9 THEN TX=0:TY=1 ELSE TX=1
:TY=0 ELSE IF AD(CX,CY+1)=9 THEN TX=1:TY=0 ELSE TX=0:TY=1 ELSE 810
800 GOTO 860
810 IF TX>0 AND TY<0 THEN IF WB>0 THEN IF AD(CX+1,CY)=9 THEN TX=0:TY=-1 ELSE TX=
1:TY=0 ELSE IF AD(CX,CY-1)=9 THEN TX=1:TY=0 ELSE TX=0:TY=-1 ELSE 830
820 GOTO 860
830 IF TX<0 AND TY>0 THEN IF WB>0 THEN IF AD(CX-1,CY)=9 THEN TX=0:TY=1 ELSE TX=-
1:TY=0 ELSE IF AD(CX,CY+1)=9 THEN TX=-1:TY=0 ELSE TX=0:TY=1 ELSE 850
840 GOTO 860
850 IF WB>0 THEN IF AD(CX-1,CY)=9 THEN TX=0:TY=-1 ELSE TX=-1:TY=0 ELSE IF AD(CX,
CY-1)=9 THEN TX=-1:TY=0 ELSE TX=0:TY=-1
860 TX=CX+TX:TY=CY+TY
870 IF AD(TX,TY)=1 THEN IF RND(1)<ZZ THEN PUT SPRITE 0,(CX*8,CY*8),6,0:RETURN EL
SE AD(TX,TY)=0:LOCATE TX,TY:PRINT " ";:PLAY "l64o2f":SC=SC-10:LOCATE 0,23:PRINT
USING "○○○ SCORE ##### ○○○";SC;
880 CX=TX:CY=TY
890 PUT SPRITE 0,(CX*8,CY*8),8,0
900 RETURN
910 '------ ﾄﾘｶﾞ ﾎﾞﾀﾝ ﾉ ﾜﾘｺﾐｼｮﾘ -------
920 IF AD(SX,SY)=1 THEN RETURN
930 LOCATE SX,SY:PRINT "◆";:PLAY "L64O7D"
940 AD(SX,SY)=1
950 SC=SC+10:LOCATE 0,23:PRINT USING "●●● SCORE ##### ●●●";SC;
960 RETURN 110
970 '------ ｼｮｳﾄﾂ ﾉ ﾜﾘｺﾐｼｮﾘ -----------
```

```
980 STRIG(1) OFF:COLOR 1,14,6:PLAY "L4004BAGFEDCD"
990 FOR I=1 TO 24:FOR J=1 TO 10:NEXT J:PRINT:NEXT I
1000 COLOR 15,1,7
1010 LOCATE 8,10:PRINT "G A M E  O V E R"
1020 LOCATE 6,13:PRINT USING "●●● SCORE ##### ●●●";SC;
1030 LOCATE 0,0
1040 END
1050 '------ ﾆﾝｹﾞﾝ ﾄ ｲﾝﾍﾞｰﾀﾞ ﾉ ｻｸｾｲ ------
1060 FOR I=0 TO 1
1070  S$=""
1080  FOR J=1 TO 8
1090   READ A$:S$=S$+CHR$(VAL("&H"+A$))
1100  NEXT J
1110  SPRITE$(I)=S$
1120 NEXT I
1130 '----- ﾆﾝｹﾞﾝ ﾄ ｲﾝﾍﾞｰﾀﾞ ﾉ ﾃﾞｰﾀ -----
1140 DATA 18,3C,DB,3C,18,3C,42,81,11,3A,14,38,50,98,24.48
1150 RETURN
```

逃亡ゲームPART II

前作の「逃亡ゲーム」を修正して、ゲームの内容を変えてみました。インベーダのほかにメロンとバナナがあらわれ、それをとるとスコアが大幅にアップ。1500点で成功！　となります。ただし、スコアがあがるにつれて、インベーダの動きも速くなりますよ。

遊び方

◉ゲームの進め方

最初の画面で、敵のパワーをキーインするところは前作と同じ。ただし、画面にはメロンとバナナがあらわれる。

スタートすると、画面は黒地に白となり、人間、インベーダのほかに、メロンとバナナが道のどこかに配置される。どちらも、これをとれば500点プラス。バリケードのほうは、前作と同じく10点だ。

◉ゲームの成功と終了

バリケードを落としながら、追いかけてくるインベーダから逃げまわる。メロンとバナナは、確実にとる。こうして、スコアが1500点になれば成功、画面が変わって「CONGRATULATION（コングラチュレイション）」つまり、オメデトウと表示される。それまでにインベーダにつかまれば、もちろんゲームオーバーだ。

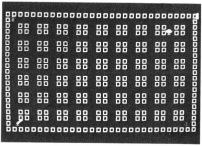
◉インベーダのパワーアップ

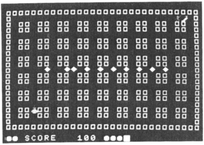
このPART IIでは、次の三つのケースでインベーダのパワーがアップするようにくふうされている。

①インベーダがバリケードを食うと、スコアが20点ずつマイナスされる。



①最初の画面


②ゲーム画面



③果物をとる



④成功！

②インベーダがメロンかバナナを食うと、追いかけるスピードがアップする。
③スコアが1000点を超えると、インベーダのスピードが極端にアップする。

ゲームのコツと注意

◎メロンとバナナをインベーダに食われないこと。インベーダが追いかけてくる道にそれらがあると食われてしまうから、逃げ方を考えること。
◎メロンかバナナを食った時点でスコアが1500点を超えると、ゲーム終了とならず、そのまま続行される。これ、まさに裏ワザ、最高点をめざしたい人は、チャレンジしてみよう。



プログラムの解説

プログラムで、行番号が緑色になっているところは、前作の一部を変更したか、その行のすべてを挿入または追加した。主な部分を解説しておく。

▶最初の画面にメロンとバナナを表示する

行230〜250　メロンとバナナのスプライト・パターンを、最初の画面に表示する。そのため、行190〜220で、人間とインベーダの表示位置を前作より上にしている。

▶ゲーム画面の変更

行280　メロンとバナナを見やすくするために、バックを黒、道を白にする。

行520　インベーダがパワーアップしたときに色を変えるため、インベーダのカラー番号を変数にする。

行530〜590　メロンとバナナを、ランダムな場所で、重ならず、必ず道に表示するための座標の計算。

行620〜630　メロンとバナナを、ゲーム画面に表示する。

▶スプライト・パターンの衝突処理

行1100　2回以上の割りこみを禁止する。

行1110〜1150　どのスプライト・パターンとどのスプライト・パターンが衝突したかによって、割りこみ先を指定する。

▶スプライト・パターンの追加

行1240　ループの回数を4回にすることで、パターンを二つ追加する。

行1320　メロン、バナナのスプライト・パターンのデータを追加する。

▶割りこみの処理

行1350〜1380　インベーダがメロンかバナナを食ったとき、パワーをアップする。

行1390〜1430　人間がメロンかバナナをとったとき、スコアを500プラスする。

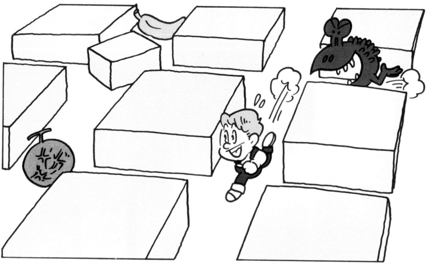
行1440　スプライトの衝突による割りこみを再び許可し、行110に戻る。

▶得点の表示と成功画面

行1460　人間がメロンかバナナをとったとき、その時点のスコア合計を表示する。

行1470　スコアが1000点を超えれば、インベーダのパワーをアップする。

行1480〜1510　スコアが1500以外なら、行1440に戻る。1500なら、音楽を鳴らし、成功の画面をつくる。



```
99 'ﾄｳﾎﾞｳ ｹﾞｰﾑ 2
100 GOSUB 150
110 GOSUB 700
120 IF RND(1)>Z/100 THEN GOSUB 830
130 GOTO 110
140 '-------- ﾒﾆｭｰ ｶﾞﾒﾝ ﾋｮｳｼﾞ ------------
150 SCREEN 1,1:WIDTH 32:COLOR 15,1,1:CLS:KEY OFF
160 DD=RND(-TIME)
170 LOCATE 8,1:PRINT "ﾄｳﾎﾞｳ ｹﾞｰﾑ 2 !!!!"
180 GOSUB 1240
190 PUT SPRITE 1,(10*8,3*8),5,1
200 LOCATE 13,4:PRINT "ｼﾞﾌﾞﾝ ﾃﾞｽ"
210 PUT SPRITE 0,(10*8,6*8),8,0
220 LOCATE 13,7:PRINT "ﾃｷ ﾃﾞｽ"
230 PUT SPRITE 2,(6*8,9*8),3,2
240 PUT SPRITE 3,(9*8,9*8),10,3
250 LOCATE 13,10:PRINT "500 ﾃﾝ!!"
260 LOCATE 6,15:INPUT "ﾃｷ ﾉ ﾊﾟﾜｰ(1-100)";Z:Z=100-Z
270 '------ ｶﾍﾞ ﾄ ﾐﾁ ﾉ ﾋｮｳｼﾞ  ----------
280 SCREEN 1,0:WIDTH 32:COLOR 15,1,1
290 DEFINT A-Y
300 ZZ=.7
310 DIM AD(30,21)
320 FOR I=1 TO 21
330  FOR J=1 TO 30
340   LOCATE J,I:PRINT "ロ":AD(J,I)=9
350  NEXT J
360 NEXT I
370 FOR I=2 TO 20
380  M=I MOD 3
390  IF M=2 THEN M=1 ELSE M=3
400  FOR J=2 TO 29 STEP M
```

```
410   LOCATE J,I:PRINT " ":AD(J,I)=0
420  NEXT J
430 NEXT I
440 RESTORE 1320:GOSUB 1240
450 '------ ｼｮｳﾄﾂ ﾉ ﾜﾘｺﾐｻｷ ------------
460 ON SPRITE GOSUB 1100
470 '------ ﾄﾘｶﾞ ﾎﾞﾀﾝ ﾉ ﾜﾘｺﾐｻｷ --------
480 ON STRIG GOSUB ,1030
490 SPRITE ON
500 STRIG(1) ON
510 '------ ﾆﾝｹﾞﾝ ﾄ ｲﾝﾍﾞｰﾀﾞ ﾄ ｷｬﾗｸﾀ ﾉ ﾋｮｳｼﾞ ---
520 CX=2:CY=2:SX=29:SY=20:IK=8
530 MX=INT(RND(2)*28+1)
540 MY=INT(RND(2)*19+1)
550 BX=INT(RND(3)*28+1)
560 BY=INT(RND(3)*19+1)
570 IF AD(MX,MY)<>0 THEN 530
580 IF AD(BX,BY)<>0 THEN 550
590 IF MX=BX AND MY=BY  THEN 530
600 PUT SPRITE 0,(CX*8,CY*8),IK,0
610 PUT SPRITE 1,(SX*8,SY*8),5,1
620 PUT SPRITE 2,(MX*8,MY*8),3,2
630 PUT SPRITE 3,(BX*8,BY*8),10,3
640 '------ ｼﾞｮｲ ｽﾃｨｯｸ ﾉ ｼｮｷｾｯﾃｲ   ----
650 DIM XV(8),YV(8)
660 XV(1)=0:XV(2)=0:XV(3)=1:XV(4)=0:XV(5)=0:XV(6)=0:XV(7)=-1:XV(8)=0
670 YV(1)=-1:YV(2)=0:YV(3)=0:YV(4)=0:YV(5)=1:YV(6)=0:YV(7)=0:YV(8)=0
680 RETURN
690 '------- ﾆﾝｹﾞﾝ ﾉ ｾｲｷﾞｮ ------------
700 SV=STICK(1)
710 IF SV=0 THEN RETURN
720 TX=SX+XV(SV):TY=SY+YV(SV)
730 IF TX<2 THEN TX=2
```

```
740 IF TX>29 THEN TX=29
750 IF TY<2 THEN TY=2
760 IF TY>20 THEN TY=20
770 IF AD(TX,TY)=1 THEN RETURN
780 IF AD(TX,TY)=9 THEN RETURN
790 SX=TX:SY=TY
800 PUT SPRITE 1,(SX*8,SY*8),5,1
810 RETURN
820 '------- ｲﾝﾍﾞｰﾀﾞ ﾉ ｾｲｷﾞｮ ----------
830 TX=SX-CX:TY=SY-CY
840 WX=ABS(TX):WY=ABS(TY):WB=SGN(WX-WY)
850 IF WX>0 AND WY>0 THEN 900 ELSE IF WX>0 AND WY=0 THEN 880
860 IF AD(CX,CY-1)=9 AND AD(CX,CY+1)=9 THEN IF TY>0 THEN IF AD(CX-1,CY+1)=9 THEN
TX=1:TY=0 ELSE TX=-1:TY=0 ELSE IF AD(CX-1,CY-1)=9 THEN TX=1:TY=0 ELSE TX=-1:TY=0
 ELSE IF TY>0 THEN TX=0:TY=1 ELSE TX=0:TY=-1
870 GOTO 970
880 IF AD(CX-1,CY)=9 AND AD(CX+1,CY)=9 THEN IF TX>0 THEN IF AD(CX+1,CY+1)=9 THEN
 TX=0:TY=-1 ELSE TX=0:TY=1 ELSE IF AD(CX-1,CY+1)=9 THEN TX=0:TY=-1 ELSE TX=0:TY=
1 ELSE IF TX>0 THEN TX=1:TY=0 ELSE TX=-1:TY=0
890 GOTO 970
900 IF TX>0 AND TY>0 THEN IF WB>0 THEN IF AD(CX+1,CY)=9 THEN TX=0:TY=1 ELSE TX=1
:TY=0 ELSE IF AD(CX,CY+1)=9 THEN TX=1:TY=0 ELSE TX=0:TY=1 ELSE 920
910 GOTO 970
920 IF TX>0 AND TY<0 THEN IF WB>0 THEN IF AD(CX+1,CY)=9 THEN TX=0:TY=-1 ELSE TX=
1:TY=0 ELSE IF AD(CX,CY-1)=9 THEN TX=1:TY=0 ELSE TX=0:TY=-1 ELSE 940
930 GOTO 970
940 IF TX<0 AND TY>0 THEN IF WB>0 THEN IF AD(CX-1,CY)=9 THEN TX=0:TY=1 ELSE TX=-
1:TY=0 ELSE IF AD(CX,CY+1)=9 THEN TX=-1:TY=0 ELSE TX=0:TY=1 ELSE 960
950 GOTO 970
960 IF WB>0 THEN IF AD(CX-1,CY)=9 THEN TX=0:TY=-1 ELSE TX=-1:TY=0 ELSE IF AD(CX,
CY-1)=9 THEN TX=-1:TY=0 ELSE TX=0:TY=-1
970 TX=CX+TX:TY=CY+TY
980 IF AD(TX,TY)=1 THEN IF RND(1)<ZZ THEN PUT SPRITE 0,(CX*8,CY*8),6,0:RETURN EL
```

```
SE AD(TX,TY)=0:LOCATE TX,TY:PRINT " ";:PLAY "V10164o2f":SC=SC-20:LOCATE 0,23:PRI
NT USING "○○○ SCORE ##### ○○○";SC;
990 CX=TX:CY=TY
1000 PUT SPRITE 0,(CX*8,CY*8),IK,0
1010 RETURN
1020 '------ トリガ゛ ボ゛タン ノ ワリコミショリ -------
1030 IF AD(SX,SY)=1 THEN RETURN
1040 LOCATE SX,SY:PRINT "◆";:PLAY "L64O7D"
1050 AD(SX,SY)=1
1060 SC=SC+10:IF SC=1500 THEN GOTO 1480
1070 LOCATE 0,23:PRINT USING "●●● SCORE ##### ●●●";SC;
1080 RETURN 110
1090 '------ ショウトツ ノ ワリコミショリ -----------
1100 SPRITE OFF:STRIG(1) OFF
1110 IF SX=CX AND SY=CY THEN GOTO 1160
1120 IF CX=MX AND CY=MY THEN GOTO 1350
1130 IF CX=BX AND CY=BY THEN GOTO 1370
1140 IF SX=MX AND SY=MY THEN GOTO 1390
1150 IF BX=SX AND BY=SY THEN GOTO 1410
1160 STRIG(1) OFF:COLOR 1,14,6:PLAY "L40O4BAGFEDCD"
1170 FOR I=1 TO 24:FOR J=1 TO 10:NEXT J:PRINT:NEXT I
1180 COLOR 15,1,7
1190 LOCATE 8,10:PRINT "G A M E  O V E R"
1200 LOCATE 6,13:PRINT USING "●●● SCORE ##### ●●●";SC;
1210 LOCATE 0,0
1220 END
1230 '------  キャラクタ ノ サクセイ -----
1240  FOR I=0 TO 3
1250   S$=""
1260   FOR J=1 TO 8
1270    READ A$:S$=S$+CHR$(VAL("&H"+A$))
1280   NEXT J
1290   SPRITE$(I)=S$
```

```
1300  NEXT I
1310 '----- キャラクタ ノ データ -----
1320  DATA 18,3C,DB,3C,18,3C,42,81,11,3A,14,38,50,98,24,48,3C,08,3C,7E,FF,FF,7F,
3C,07,02,07,0F,1E,3C,F8,F0
1330 RETURN
1340 '----- ワリコミ ショリ -------
1350 MY=209:Z=Z/2:IK=9
1360 PUT SPRITE 2,(MX*8,MY*8),0,2:GOTO 1440
1370 BY=209:Z=Z/2:IK=9
1380 PUT SPRITE 3,(BX*8,BY*8),0,3:GOTO 1440
1390 MY=209
1400 PUT SPRITE 2,(MX*8,MY*8),0,2:GOTO 1430
1410 BY=209
1420 PUT SPRITE 3,(BX*8,BY*8),0,3:GOTO 1430
1430 SC=SC+500:GOSUB 1460
1440 SPRITE ON:STRIG(1) ON:GOTO 110
1450 '----- トクテン ノ ヒョウジ------
1460 LOCATE 0,23:PRINT USING "●●● SCORE ##### ●●●";SC;
1470 IF SC>1000 THEN Z=Z/10
1480 IF SC<>1500 THEN RETURN ELSE PLAY"V10O4L16CR32CR32CDR32DR32DL2E":CLS:COLOR
1,15,5
1490 LOCATE 8,10:PRINT "CONGRATULATION!!"
1500 LOCATE 6,13:PRINT USING "●●● SCORE ##### ●●●";SC;
1510 END
```

すごろくゲーム

このゲームは、追いかけてくるパクパクマンに追いつかれないようにしながら画面のコマを進め、ゴールをめざすもの。ふつうのすごろくは、サイコロで進む数が決まりますが、この場合、ゲームの決め手になるのは計算能力です。

まちがいが多ければ、たちどころにゲームオーバー。すばやく答えを出し、ゴールをめざしてください。



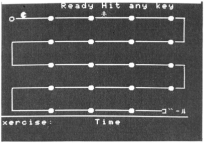
①開始
F5/F10
⬇
② ? 何かキーを押す
⬇
③ゲーム
1 ～ 0
＋
RETURN
［　　　］答えの取り消し
⬇
④ゴール／ゲームオーバー
⬇
⑤終了

遊び方

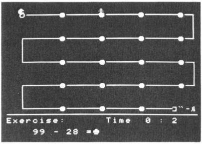
●ゲームの進め方

実行すると、すごろく画面があらわれ、「Ready Hit any key（レディ ヒット エニー キー）」という表示が出る。準備ができたら何かキーを押してください、という意味なので、どれでもよいから一つキーを押す。これで、ゲームスタートとなる。

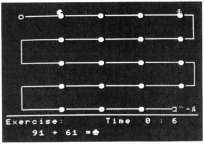
スタートすると、画面の左下に問題が表示される。２桁（けた）の足し算か引き算だか



①最初の画面



②問題の表示



③3秒以内で正解(2コマ進む)

ら、すばやく計算し、解答をキーインしてから RETURN キーを押す。

キミの解答が正しければ、軽快な音楽が流れ、すごろくのコマである画面の人間が進む。解答がまちがっていたり、制限時間(10秒)内に答えられなかったら、ブーとブザーが鳴り、パクパクマンだけが進む。

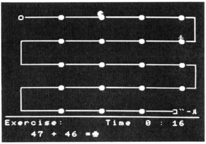
こうしてゲームを続け、キミのコマがゴールインすれば成功。それまでにかかった時間とともに、「オメデトウ」が表示される。

まちがいが続いたり、解答に時間がかかったりして、パクパクマンに追いつかれると、その時点でゲームオーバー。ザンネン画面が表示されて終了となる。

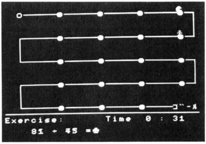
●コマとパクパクマンの動き

キミのコマは、解答までにかかった時間によって、動き方が違う。問題が出て、3秒以内に正解したら2コマ、それを超えて、10秒以内なら1コマ先へ進む。

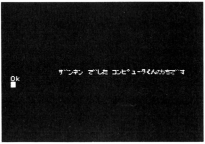
一方、パクパクマンのほうは、キミの解答にかかわらず、1問につき1コマずつ進む。



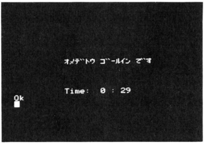
④4〜10秒で正解(1コマ進む)



⑤不正解(1コマ追いつかれる)



⑥ゲームオーバー



⑦成功、ゴールイン!

●解答の訂正

答えをキーインしてからまちがいに気づいたら、すぐに［　　］キーを押そう。キミの入力した数字が消えて、もう一度、解答できる状態になる。

ただし、RETURN キーを押してしまってからでは訂正がきかないから、その点に注意。

10秒の制限時間内なら、何度でも訂正することができる。

ゲームのコツと注意

◎前述したように、解答は訂正できるから、あやしいと思ったら、答えを確認してから RETURN キーを押すこと。

◎出題は、引き算より足し算のほうが多い。比較的やさしい足し算で、できるだけコマを進めるのがコツだ。

◎簡単にゴールインできる人は、タイムを競ってみよう。30秒以内でゴールインなら、まあ水準以上といえる。

プログラムの解説

初期設定〈行100〜120〉

行100　使用する画面をテキスト・モードにし、ファンクション・キーの表示を消し、使う色を決める。

行110　すべての変数を、整数型に宣言する。

行120　乱数の発生が、実行のたびに変わるようにする。

各処理の呼び出しと実行〈行130〜170〉

行130　行190へ飛んで、スプライトをセットする。

行140　行310へ飛んで、画面を表示する。

行150　行590へ飛んで、ゲームを実行する。

行160　行1140へ飛んで、ゲーム終了の処理をする。

スプライト・パターンのセット〈行180〜290〉

行190〜240　スプライト・パターンのデータ読みこみのループ。パクパクマン、人間をつくる。

行250　スプライト番号0に、パクパクマンをセットする。

行260　スプライト番号1に、人間をセットする。

行280〜290　スプライト・パターンのデータ。

画面の表示〈行300〜400〉

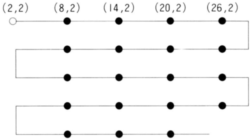
行310〜320　○の位置の初期値をセットして、画面に表示する。

行330　すごろく画面の表示処理（行420）を呼び出して、実行する。

行340〜370　すごろく画面のまわりの文字や線の表示。

行380〜390　パクパクマンと人間の最初の位置を決め、表示する。

すごろく画面の処理〈行410〜570〉



行420〜550　画面の形と座標は、図のとおり。最初の○は、行310〜320で表示している。行420〜450で、横線と●を描き、行460〜530で折り返し部分の ］［ を表示する。

行560　人間のスプライト・パターンを表示する。

行570　パクパクマンのスプライト・パターンを表示する。

ゲームの実行〈行580〜730〉

行590〜630　ゲーム開始の指示を表示し、各変数の初期値を設定する。

行640　行850を呼び出して、計算式を表示する。

行650〜670　経過時間を、秒単位で表示する。

行680　A＄にキーインされた解答を入れる。

行690　解答までに10秒間かかったかどうかをチェックし、超えていたら強制的に RETURN キーが押された状態にする。

行700　解答がキーインされたら、判定の処理（行750）を呼び出して実行する。

行710　パクパクマンが人間に追いついたかどうかを判断し、追いついていたらEFに1をセットし、ゲームの処理を終わる。

行720　人間がゴールに到着したかどうかを判断し、ゴールしていたらEFに2をセットし、ゲームの処理を終わる。

行730　次の問題を表示するかどうかの判断。表示するなら行630、しないのなら行650に戻る。

キーインされた解答の処理〈行740〜830〉

行750　RETURN キーが押されたかどうかを判断し、押されていたらそれまでにかかった時間をセットし、正誤の判定処理をして解答の処理を終わる。

行760　[　　　] キーが押されたかどうかを判断し、押されていたら表示された解答を消して行820（♣の表示）へ飛ぶ。

行770　解答の数字が3桁(けた)を超えた場合、ブザーを鳴らし、解答の処理を終わる。

行780　キーインされた内容が数字の0～9かどうかを判定し、それ以外の場合は解答の処理を終わる。

行790～810　キーインされた数字を解答欄に表示し、解答をチェックするための文字列にする。

行820　♣を、解答入力欄の次の桁(けた)に表示する。

▶問題の作成と解答のチェック〈行840～970〉

行850　各変数の初期値をセットする。

行860～870　10～99の乱数を二つ発生させ、EA、EBに代入する。

行880　EAの値がEBより小さいとき、行1290へ飛んで足し算を出題する。

行890～900　1か2の乱数を発生させ、SLに入れる。SLが1なら引き算(行1260)、それ以外なら足し算（行1290）の出題処理へ飛ぶ。

行910　キーインされた解答をチェックし、誤っていれば音を出し、行960へ飛ぶ。

行920　経過時間の計算。

行930　人間のスプライト・パターンがゴール直前にあるとき、正解によって進むコマ数SLを1にする。

行940　正解数キーインまでの時間が3秒以下のとき、SLを2にする。それ以外のときは、SLを1にする。

行950　正解の場合の処理。

行960　不正解の場合の処理。

▶スプライト・パターンの移動〈行980～1050〉

行990　進めるコマ数まで処理をくり返すループの開始。

行1000　正解のとき、行1210を呼び出して音楽を鳴らす。

行1010　スプライト・パターンが右向きに動いていて、そのX座標が222（すごろく画面の右端）のとき、縦方向に動かす処理を実行する。

行1020　スプライト・パターンが左向きに動いていて、そのX座標が78（すごろく画面の左端）のとき、縦方向に動かす処理を実行する。

行1030　スプライト・パターンを水平方向に動かす処理を呼び出して、実行する。

▶縦の移動〈行1060～1090〉

行1060　動くほうのスプライト・パターンのY座標に、32をプラスする（下に動

かす)。

行1070〜1080　動いたスプライト・パターンの向きを変える(右向きなら左向きに、左向きなら右向きに)。

◆横の移動〈行1100〜1120〉

行1100〜1120　スプライト・パターンを水平方向に動かす処理。右向きならX座標に48をプラスし、左向きなら48をマイナスする。

◆ゲーム終了の処理〈行1130〜1190〉

行1140　画面を消し、新たにゲーム終了の画面を設定する。

行1150〜1160　ゴールインしたとき、途中でゲームオーバーになったとき、それぞれのメッセージを表示する。

行1170　ゴールインしたとき、所要時間を表示する。

行1180〜1190　終了画面をしばらく表示し、処理を終わる。

◆音楽の演奏と問題表示のサブルーチン〈行1200〜1310〉

行1200〜1240　正解のとき、音楽を鳴らす処理。

行1250〜1310　引き算、足し算の問題を表示して、それぞれを計算し、答えを変数ANに入れる。

使用変数

AN　出題式の答え

AP$　キーインされた解答(3桁(けた)に設定

CJ　パクパクマンの進行方向(右向き=0、左向き=1)

CM　パクパクマンの動き(動くとき=1、動かないとき=0)

CX、CY　パクパクマンの位置(X座標とY座標)

EF　ゲームの状態(人間が追いつかれたとき=1、ゴールインしたとき=2)

ES　次の問題表示の選択(表示する=1、表示しない=0)

PJ　人間の進行方向(右向き=0、左向き=1)

PL　人間の動き(動くとき=1、動かないとき=0)

PX、PY　人間の位置　(X座標とY座標)

SE　解答の正誤(正解=1、不正解=0)

SL　足し算か引き算をセットする変数

ST　スプライト・パターンの進み方(1コマ=1、2コマ=2)

T　ゲームのスタート時間

T1　経過時間

```
99  'けいさん すご゛ろく
100 SCREEN 1:KEY OFF:CLS:COLOR 15,1
110 DEFINT A-Z
120 Q=RND(-TIME)
130 GOSUB 190 'すふ゜らいと せっと
140 GOSUB 310'か゛めん ひょうし゛
150 GOSUB 590'めいん
160 GOSUB 1140'あとしょり
170 END
180 'すふ゜らいと せっと
190 FOR I=1 TO 8
200   READ C$:C1$=C1$+CHR$(VAL("&h"+C$))
210 NEXT I
220  FOR I=1 TO 8
230    READ P$:P1$=P1$+CHR$(VAL("&h"+P$))
240  NEXT I
250 SPRITE$(0)=C1$
260 SPRITE$(1)=P1$
270 RETURN
280 DATA 38,7c,fe,f8,f0,fe,7c,38
290 DATA 18,3c,18,7e,5a,18,24,42
300 'か゛めん ひょうし゛
310 GX=2:GY=2
320 LOCATE GX,GY:PRINT "○";
330 GOSUB 420
340 LOCATE 25,18:PRINT "コ゛ール"
350 LOCATE 0,19:PRINT "──────────────────────────────"
360 LOCATE 0,20:PRINT "Exercise:"
370 LOCATE 15,20:PRINT "Time"
380 CX=30:CY=10:GOSUB 570
390 PX=126:PY=10:GOSUB 560
400 RETURN
410 'すご゛ろく か゛めん
```

```
420 FOR J=1 TO 5
430   FOR GX=3 TO 26 STEP 6
440      LOCATE GX,GY:PRINT "―――――●";
450   NEXT GX
460   LOCATE 27,GY:PRINT "─"
470   IF J=5 THEN   540
480   IF (J MOD 2)=1 THEN GX=28:S$="┐":S1$="┘" ELSE GX=2:S$="┌":S1$="└"
490   LOCATE GX,GY:PRINT S$
500   FOR GY=GY+1 TO GY+2
510    LOCATE GX,GY:PRINT "│"
520   NEXT GY
530   LOCATE GX,GY:PRINT S1$
540 NEXT J
550 RETURN
560 PUT SPRITE 1,(PX,PY),2,1:RETURN
570 PUT SPRITE 0,(CX,CY),11,0:RETURN
580 'ｹﾞｰﾑ
590 LOCATE 10,0:PRINT "Ready Hit any key "
600 R$=INKEY$ :IF R$="" THEN 600
610 LOCATE 10,0:PRINT SPACE$(18);
620 TIME=0
630 HX=12
640 GOSUB 850
650 A=TIME/60-X*60
660 IF A=60 THEN B=B+1:X=X+1
670 LOCATE 20,20:PRINT B;:PRINT ":";:PRINT A
680 A$=INKEY$
690 IF T+600<TIME THEN A$=CHR$(&HD)
700 IF A$<>"" THEN GOSUB 750
710 IF PX=CX AND PY=CY THEN EF=1:RETURN
720 IF PX=222 AND PY=138  THEN EF=2:RETURN
730 IF ES=1 THEN 630 ELSE 650
740 'かいとう の にゅうりょく
```

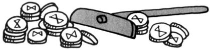
```
750 IF A$=CHR$(&HD) THEN T1=TIME:GOSUB 910:GOTO 830
760 IF A$=CHR$(&H20) THEN AP$="":HX=12:LOCATE HX+1,22:PRINT SPACE$(10):GOTO 820
770 IF LEN(AP$)=3 THEN BEEP:RETURN
780 IF A$<CHR$(&H30) OR A$>CHR$(&H39) THEN RETURN
790 HX=HX+1
800 LOCATE HX,22:PRINT A$;
810 AP$=AP$+A$
820 LOCATE HX+1,22:PRINT "♣"
830 RETURN
840 'もんだい の ひょうじ
850 T=TIME:ES=0:AP$=""
860 EA=INT(RND(1)*90+10)
870 EB=INT(RND(1)*90+10)
880 IF EA<EB THEN GOTO 1290
890 SL=INT(RND(1)*2+1)
900 IF SL=1 THEN GOTO 1260 ELSE GOTO 1290
910 IF AN<>VAL(AP$) THEN PLAY "o218"+"b","o218"+"a","o218"+"g":GOTO 960
920 T2=INT (T1-T)/60
930 IF PX=174 AND PY=138 THEN ST=1:GOTO 950
940 IF T2<=3 THEN ST=2 ELSE ST=1
950 PL=1:SE=1:CM=0:GOSUB 990
960 ES=1:SE=0:PL=0:ST=1:CM=1:GOSUB 990
970 RETURN
980  'スプライト パターン の ひょうじ
990 FOR E=1 TO ST
1000   IF SE=1 THEN GOSUB 1210
1010   IF (SE=1 AND PJ=0 AND  PX=222) OR (CM=1 AND CJ=0 AND CX=222) THEN GOSUB 1
060:GOTO 1040
1020   IF (SE=1 AND PJ=1 AND  PX=78 ) OR (CM=1 AND CJ=1 AND CX=78  ) THEN GOSUB
1060:GOTO 1040
1030   GOSUB 1100
1040 NEXT E
1050 RETURN
```

```
1060 IF PL=1 THEN PY=PY+32:GOSUB 560 ELSE CY=CY+32:GOSUB 570
1070 IF PL=1 THEN IF PJ=1 THEN PJ=0 ELSE PJ=1
1080 IF PL<>1 THEN IF CJ=1 THEN CJ=0 ELSE CJ=1
1090 RETURN
1100 IF (PL=1 AND PJ=0) OR (CM=1 AND CJ=0) THEN DH=48 ELSE DH=-48
1110 IF PL=1 THEN PX=PX+DH:GOSUB 560  ELSE CX=CX+DH:GOSUB 570
1120 RETURN
1130 'ｹﾞｰﾑ の おわり
1140 CLS:SCREEN 0:WIDTH 40 :COLOR 14,1
1150 IF EF=2 THEN MS$="ｵﾒﾃﾞﾄｳ ｺﾞｰﾙｲﾝ てﾞす" ELSE MS$="ｻﾞﾝﾈﾝ てﾞした ｺﾝﾋﾟｭｰﾀくんのかちてﾞす"
1160 LOCATE 10,10:PRINT MS$
1170 IF EF=2 THEN LOCATE 10,15:PRINT "Time: ";:PRINT B;:PRINT ":";:PRINT A
1180 FOR T=1 TO 2000:NEXT T
1190 RETURN
1200 'おと を たﾞす
1210 PLAY "l9o5e"
1220 PLAY "l9o5f"
1230 PLAY "l9o5g"
1240 RETURN
1250 'もんたﾞい の せんたく
1260 LOCATE 3,22:PRINT EA;:PRINT "-";:PRINT EB;:PRINT "=";:PRINT "♣    "
1270 AN=EA-EB
1280 RETURN
1290 LOCATE 3,22:PRINT EA;:PRINT "+";:PRINT EB;:PRINT "=";:PRINT "♣    "
1300 AN=EA+EB
1310 RETURN
```

カジノでルーレット MSX2対応

カジノで人気のあるゲームといえば、まず名前のあがるルーレット。回転する円盤の中で球を転がし、止まった位置を当てるおなじみのゲームです。

それを、パソコンで実現したのがこのゲーム。数字、色のどちらでも、キミのすきなほうでかけることができ、遊び方も簡単。500点の持ち点をどこまで増やすか、さすらいのギャンブラー気分を十分に味わってください。



①開始

F5/F10

↓

②ゲーム

A ～ R でかける

↓

Y／N かけた位置の確認

↓

RETURN スタート

↓

③終了

持ち点 0 で自動的に終了

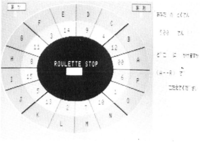
遊び方

●ゲームの進め方

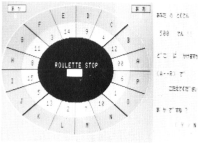
最初の持ち点は500点。かけ方は、赤、青の色で選ぶか、ズバリその数字を予測するかのどちらか。1回のかけ点は100点で、色で当たれば2倍、数字で当たれば16倍の得点が戻ってくる。

● 1 回のかけ方

実行すると、ルーレット盤が少しずつ描かれ、それと同時に右に説明画面があ



①最初の画面



②赤（Q）を選び、Yを押す

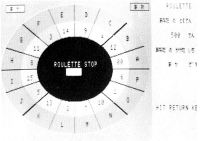
らわれる。あとは、次の手順でゲームをくり返す。

①色でいくならQ（赤）、R（青）のどちらか、数字（00～15）でいくなら、それに対応するA～Pのキーを一つ押す。

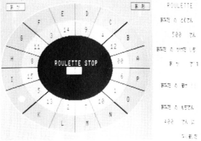
②選んだ色または数字を確認してくるから、それでOKなら[Y]、変更するなら[N]を押す。変更の場合は、もう一度①の要領でキーイン。

③説明画面が変わり、いちばん下に「HIT RETURN KEY」という文字が点滅する。RETURN キーを押せば球がスタートするので、ここだと思うところで押す。

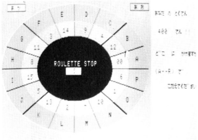
④球が回転し、しばらくして止まる。



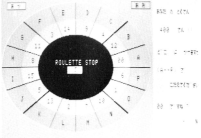
③ RETURN キーで球がスタート



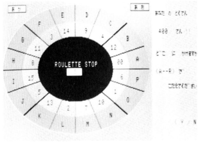
④球がストップ



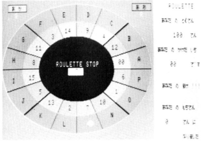
⑤次のゲーム画面



⑥00（[A]）を選択



⑦変更なら[N]をキーイン



⑧0点

⑤その結果によって持ち点が変わり、①の説明画面になる。次の色または数字を選ぶ。

●ゲームの終了

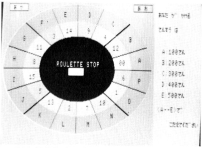
持ち点が0になれば、しばらくその結果が表示されてゲームが終わる。

途中でやめたいときは、**CTRL** キーを押しながら **STOP** キーを押す。

ゲームの改造

◘かけ点を自由に選べるようにする

以下のように修正すれば、1回のかけ点を100～500点の間から自分で選べるようになる。



▲かけ点の入力

①行640～650、行670～680の「100」を、「MT」(かけ点を入れる変数)に変える。

②行990のあとに、「GOSUB　1430」を追加する(かけ点を自由に選べる処理を呼び出す)。

③行1360を修正し、新たに行1361～1362を次のように挿入する。

```
1360 COLOR 1,11,4:PRESET(470,90):PRINT #1,"で"
1361 COLOR 9,11,4:PRESET(420,110):PRINT #1,MT
1362 COLOR 1,11,4:PRESET(470,110):PRINT #1,"てん です!"
```

④行1430～1590を、次のように追加する。

```
1430 '  *** ﾃﾝｽｳ ﾆｭｰﾘｮｸ ｶﾞﾒﾝ ***
1440 COLOR 9,11,4:PRESET(390,20):PRINT #1,"あなた が  かける"
1450              PRESET(390,40):PRINT #1,"てんすう は"
1460 COLOR 9,11,4:PRESET(400,70):PRINT #1,"A:100てん"
1470              PRESET(400,85):PRINT #1,"B:200てん"
1480              PRESET(400,100):PRINT #1,"C:300てん"
1490              PRESET(400,115):PRINT #1,"D:400てん"
1500              PRESET(400,130):PRINT #1,"E:500てん"
1510 COLOR 5,11,4:PRESET(390,150):PRINT #1,"(A--E)で"
1520              PRESET(420,170):PRINT #1,"こたえてください"
1530 C$=INKEY$:IF C$="" OR (C$<"A" OR C$>"E") THEN 1530
1540 IT=ASC(C$)
1550 IF IT=>65 AND IT<=69 THEN MT=(IT-64)*100:GOTO 1580
1560 IF IT=>97 AND IT<=101 THEN MT=(IT-96)*100:GOTO 1580
1570 GOTO 1530
1580 IF MT>TT THEN 1530
1590 RETURN
```

持ち点、当たったときの配点を変える

得点に関するプログラムは、以下のようになっている。これらを、すきなように修正すればよい。

①最初の持ち点は、行120の「TT＝500」で設定されている。

②各配点は、配列変数KAKE（18）に収められ、行280のDATE文で与えられている。

③配点の計算式は、行640（赤の場合）、行650（青の場合）、行670（数字の場合）で設定されている。

④はずれた場合の減点は、行680で設定されている。

プログラムの解説

初期設定〈行100〜300〉

行100〜130　すべての変数を、倍精度実数型にする。使用する画面のモード、表示する色、各変数の初期値を決め、配列を宣言する。

行140〜160　ルーレット上の数字を配列にセットするループ。

行170〜190　アタリのときの点数の加算率をセットするループ。

行200〜220　ルーレット上の文字を配列にセットするループ。

行230〜260　球のスプライト・パターンを定義するループ。

行270〜300　表示用の数字、加算率、文字、スプライト・パターンの各データ。

ゲーム画面の作成〈行310〜540〉

行310　球が移動する座標を計算する。

行320〜340　ルーレット盤をつくり、色をぬる。

行350〜370　盤の上方左右に、赤と青をあらわす四角形を表示する。

行380　行1020へ飛んで、盤のりんかくをつくる処理を実行する。

行390　区分けされた盤を、赤と青にぬり分ける。

行400　ルーレットの文字を表示する。

行410〜420　赤と青の四角に、QとRを表示する。

行430　ルーレット上の数字を表示する。

行440〜450　赤と青の四角に「あか」「あお」と表示する。

行460〜530　ルーレット盤の円の再表示と、真ん中に「ROULETTE STOP」の文字を表示する。

行540　ゲーム進行に伴う画面の表示。

▶ゲームの開始〈行550〜760〉

行550〜630　ルーレット上の球を移動させる。

行640〜680　アタリかどうかを判定し、得点の加算率を与える。

行690〜750　各ゲームの結果と、持ち点の表示。

行760　持ち点が0になったかどうかを判断する。持ち点が残っていれば、行540に戻って次のゲーム画面に。0以下なら、プログラムの実行を終わる。

▶各サブルーチン〈行770〜1420〉

行770〜1000　どこにかけるかを決めるための画面を表示する。

行1010〜1070　ルーレットのりんかくを表示する。

行1080〜1220　ルーレットの表示。

行1230〜1280　説明画面を消す。

行1290〜1420　持ち点、かけた位置を表示して、RETURN キーを押すとスタートする説明画面を表示する。

使用変数

BP　球の移動するポイントのための配列

I　球が停止した位置を示すカウンタ

KI　ルーレットのX軸の補正率

K2　ルーレットのY軸の補正率

KAI　球が移動する回数

KAKE　アタリのときの点数加算率

KP$　ルーレット上の文字。球の位置と画面上の数字、文字は、下表のように対応している

球の位置	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
数　字	00	12	4	9	14	3	11	8	15	5	13	2	7	10	1	6
文　字	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P

MP　かけた球の位置

NUM$　ルーレット上の数字

PI　円周率

R　ルーレットの半径

RU　球がスタートするポイント

S1　ルーレットをあらわす円の中心のX座標

S2　ルーレットをあらわす円の中心のY座標

SF　ルーレットの球の位置、赤と青の色分け、数学と文字の書き分けを行うためのスイッチ

SP　ルーレットの各マス目の個数

```
99 ' ROULETTE
100 DEFDBL A-Z : SCREEN 7,0 : COLOR 15,4,4 : CLS
110 OPEN "GRP:" FOR OUTPUT AS #1
120 GR=0 : TT = 500 : D$ = "" : RU = 0
130 DIM BP(2,16),NUM$(18),KAKE(18),KP$(18) : PI = 3.14159265#
140 FOR I=1 TO 18
150   READ A$ : NUM$(I) = A$
160 NEXT I
170 FOR I=1 TO 18
180   READ A : KAKE(I) = A
190 NEXT I
200 FOR I=1 TO 18
210   READ B$ : KP$(I) = B$
220 NEXT I
230 FOR I=1 TO 8
240   READ D$:E$=E$+CHR$(VAL("&H"+D$))
250 NEXT I
260 SPRITE$(1)=E$
270 DATA "00","12"," 4"," 9","14"," 3","11"," 8","15"," 5","13"," 2"," 7","10","
 1"," 6","あか","あお"
280 DATA 16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,1,1
290 DATA "A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R"
300 DATA 3C,7E,FF,FF,FF,FF,7E,3C
310 GR = 0:SF = 1:SP = PI/8:S1 = 95:S2 = 95 :R = 155:K1 = .5:K2 = .5:GOSUB 1090
320 CIRCLE (195,100),179,7 : PAINT (200,100),2,7
330 CIRCLE (195,100),130,1 : PAINT (195,100),5,1
340 CIRCLE (195,100),90,1  : PAINT (195,100),1,1
350 LINE (30,10)-(75,20),1,B
360 PAINT (31,11),9,1
370 LINE (330,10)-(375,20),1,B
380 GOSUB 1020
390 GR = 0:SF = 2:SP = PI/4:S1 = 195:S2 = 100:R = 120:K1 = 1:K2 = .5:GOSUB 1090
```

```
400 GR = 0:SF = 3:SP = PI/8:S1 = 193:S2 = 98 :R = 153:K1 = 1:K2 = .5:GOSUB 1090
410 COLOR 15,4,4 : PRESET (48,22)  : PRINT #1,KP$(17)
420 COLOR 15,4,4 : PRESET (350,22) : PRINT #1,KP$(18)
430 GR = 0:SF = 4:SP = PI/8:S1 = 190:S2 = 98 :R = 110:K1 = 1:K2 = .5:GOSUB 1090
440 COLOR 5,9,4 : PRESET (40,12)  : PRINT #1,NUM$(17)
450 COLOR 9,4,4 : PRESET (340,12) : PRINT #1,NUM$(18)
460 LINE (380,0)-(380,250),2
470 PAINT (390,10),11,2
480 CIRCLE (195,100),179,7
490 CIRCLE (195,100),130,9
500 CIRCLE (195,100),90,6
510 PAINT (195,110),1,6
520 COLOR 15,1,4 : PRESET (147,90) : PRINT #1,"ROULETTE STOP"
530 LINE (175,99)-(210,108),15,BF
540 GOSUB 780 : GOSUB 1300 : GR = 0 : I = RU
550 IF I = 16 THEN I = 1
560 I = I + 1 : GR = GR + 1
570 PLAY "L64V5B","L64V9C"
580 PUT SPRITE1,(BP(1,I),BP(2,I)),9,1
590 COLOR 8,15,4 : PRESET (185,100) : PRINT #1,NUM$(I)
600 FOR J = 0 TO 30:NEXT J
610 IF GR>KAI THEN RU = I : GOTO 640
620 IF I = 16 THEN I = 0
630 GOTO 550
640 IF MP = 17 AND (I MOD 2) = 1 THEN TT = TT + 100 * KAKE(MP) : GOTO 690
650 IF MP = 18 AND (I MOD 2) = 0 THEN TT = TT + 100 * KAKE(MP) : GOTO 690
660 ATARI = (I MOD 16)
670 IF ATARI = MP THEN TT = TT + 100 * KAKE(I) : GOTO 690
680 TT = TT - 100 : GOTO 700
690 COLOR 5,11,4:PRESET (400,120):PRINT #1,"あなた の かち !!!" : GOTO 710
700 COLOR 9,11,4:PRESET (400,120):PRINT #1,"あなた の まけ !!!"
710 COLOR 1,11,4:PRESET (400,150):PRINT #1,"あなた の もちてん"
720 COLOR 9,11,4:PRESET (400,170):PRINT #1,TT
```

```
730 COLOR 1,11,4:PRESET (450,170):PRINT #1,"てん に"
740 COLOR 1,11,4:PRESET (450,190):PRINT #1,"なりました !"
750 FOR I=1 TO 3000 : NEXT I : GOSUB 1240 : COLOR 0,4,4 : PUT SPRITE1,(520,210),
0,0
760 IF TT > 0 THEN 540 ELSE COLOR 15,4,7 : END
770 '                                          ***  BOX ノ リンカク     ***
780 KAI = RND(-TIME) * 100 +5
790 COLOR 5,11,4:PRESET (390,20):PRINT #1,"あなた の とくてん"
800 COLOR 9,11,4:PRESET (390,45):PRINT #1,TT
810 COLOR 5,11,4:PRESET (440,45):PRINT #1,"てん !!"
820              PRESET (390,75):PRINT #1,"ど゛こ  に  かけますか ?"
830              PRESET (390,100) : PRINT #1,"(A--R) で゛"
840              PRESET (420,120) : PRINT #1,"こたえてくだ゛さい"
850 B$ = INKEY$ : IF B$="" OR (B$<"A" OR B$>"r") THEN 850
860 B = ASC(B$)
870 IF B = 81 OR B = 113 THEN MP = 17 : GOTO 920
880 IF B = 82 OR B = 114 THEN MP = 18 : GOTO 920
890 IF B => 65 AND B<=80 THEN MP = B - 64  : GOTO 920
900 IF B => 97 AND B<=112 THEN MP = B - 96 : GOTO 920
910 GOTO 850
920 COLOR 12,11,4:PRESET (390,150):PRINT #1,NUM$(MP)
930 COLOR 5,11,4 :PRESET (420,150):PRINT #1,"で゛すね ?"
940               PRESET (435,170):PRINT #1,"( Y / N )"
950 G$ = INKEY$ : IF G$="" THEN 950
960 IF G$="Y" OR G$="y" THEN 990
970 IF G$="N" OR G$="n" THEN COLOR 0,11,4:PRESET (390,150):PRINT #1,"
   ":GOTO 790
980 GOTO 950
990 GOSUB 1240
1000 RETURN
1010 '                                          ***  ルーレット ノ リンカク  ***
1020 FOR I=1 TO 16
1030   L= 2*I/16
```

```
1040   CIRCLE(195,100),179,1,-1*K*3.14159265#,-1*L*3.14159265#
1050   K=L
1060 NEXT I
1070 RETURN
1080 '                                        *** ﾙｰﾚｯﾄ ﾉ ｹｲｻﾝ  ***
1090 FOR I=PI/16 TO PI*2 STEP SP
1100   PX = S1 + R * COS(I) * K1
1110   PY = S2 - R * SIN(I) * K2
1120   GR=GR+1
1130   IF SF = 1 THEN BP(1,GR) = PX : BP(2,GR) = PY : IF GR = 16 THEN 1180
1140   IF SF = 2 THEN PAINT (PX,PY),9,1
1150   IF SF = 3 THEN COLOR 1,2,4:PRESET (PX,PY):PRINT #1,KP$(GR):IF GR>=16 THEN
 1180
1160   IF SF = 4 THEN GOSUB 1200
1170 NEXT I
1180 RETURN
1190 '                                        *** ﾙｰﾚｯﾄ ﾉ ｽｳﾁ    ***
1200 IF (GR MOD 2) = 1 THEN COLOR 1,9,4 : PRESET (PX,PY) : PRINT #1,NUM$(GR)
1210 IF (GR MOD 2) = 0 THEN COLOR 1,5,4 : PRESET (PX,PY) : PRINT #1,NUM$(GR)
1220 RETURN
1230 '                                        *** ｶﾞﾒﾝ ﾐｷﾞ ｸﾘｱ   ***
1240 COLOR 0,11,4
1250 FOR I=0 TO 200 STEP 5
1260   PRESET (380,I) : PRINT #1,"                "
1270 NEXT I
1280 RETURN
1290 '                                        *** ｶﾞﾒﾝ-2 ﾉ ｻｸｾｲ  ***
1300 COLOR 1,11,4:PRESET(410,10):PRINT #1," ROULETTE "
1310 COLOR 1,11,4 : PRESET (400,30) : PRINT #1,"あなた の とくてん"
1320 COLOR 9,11,4 : PRESET (420,50) : PRINT #1,TT
1330 COLOR 1,11,4 : PRESET (470,50) : PRINT #1,"てん !!"
1340 COLOR 1,11,4 : PRESET (390,70) : PRINT #1,"あなた の かけた いち は"
1350 COLOR 9,11,4 : PRESET (420,90) : PRINT #1,NUM$(MP)
```

```
1360 COLOR 1,11,4 : PRESET (470,90) : PRINT #1,"ﾃﾞｽ !"
1370 COLOR 9,11,4 : PRESET (390,150): PRINT #1,"HIT RETURN KEY"
1380 Y$ = INKEY$
1390 COLOR 0,11,4 : PRESET (390,150): PRINT #1,"              "
1400 IF Y$ <> CHR$(13) THEN 1370
1410 COLOR 0,11,4 : PRESET (390,150): PRINT #1,"              "
1420 RETURN
```

電話ファイル

転居、改姓など、何かと書きなおしの多いのが、電話帳です。また、急な用件で連絡をとりたいのに、かんじんの電話帳が見あたらない、というのもよくあることです。

そんなときに備えて、自分だけの元帳を、パソコン電話ファイルでつくっておきましょう。登録件数は50件、変更は自由にできますし、50音順の一覧表で、目的の電話番号をすぐに探すこともできます。

なお、この電話ファイルには、フロッピィディスクが必要です（カセットテープでは使えない)。また、RAM（ラム）容量が64キロバイト以上の機種でないと、大量のデータを入れたときに処理しきれないことがあります。

①開始
F5/F10
↓
②ゲーム
1 登録
2 変更
3 一覧
↓
③終了
9

遊び方

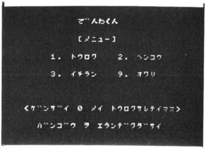
●フロッピィディスクのセット

まず、初期化した（マニュアル参照）フロッピィディスクを、ディスク装置にセットする。ディスクは、ほかのプログラム保存用のものをそのまま使うこともできるが、電話ファイルだけのプログラムとデータ保存用のものをつくっておくほうがいいだろう。

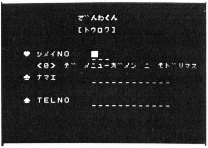
●新規登録

実行すると、最初のメニュー画面があらわれ、処理選択の説明と現在の登録数が表示される。処理は、1が登録、2が変更、3が一覧、9が終了で、それぞれ1～3と9のキーに対応している。

最初は1をキーインして、登録の画面になる。ここでの入力のしかたは、次のとおり。



①最初のメニュー画面



②登録画面



③入力終了

①氏名番号の入力。3桁(けた)までの数字をキーインする。とりあえずは1だが、1を入力しても01や001を入力しても、パソコン内部では同様に扱われる。

すでに登録ずみの番号を入力してしまったら、ブザー音とそれを示すメッセージが出るから、一覧処理で確認して、登録されていない番号を入力する。ここでは[0][0][1]とキーインして[RETURN]キーを押そう。

②画面のカーソルが次の行に移って、氏名の入力になる。アルファベット、ひらがな、カタカナのどれで入力してもよい(グラフィック・シンボルを使うこともできる)。ただし、登録できるのは12文字まで(カナなら濁点も1字ぶん)。アルファベットは長くなるので、ここではカタカナで入力してみよう([CAPS LOCK]、[かな]キーをロック)。終われば、[RETURN]キーを押す。

③電話番号の入力。終われば、[RETURN]キーを押す。

④入力がすべて終わったら、登録するかどうかを聞いてくるので、登録するなら[Y][RETURN]、しないなら[N][RETURN]とキーインする([かな]キーのロックを解除しておく。以下同じ)。どちらの場合も、画面のデータが消えて、新しい登録画面になる。

⑤続けて登録するなら、①〜④の手順をくり返す。登録を終わる場合は、氏名番号の入力のところで[0][RETURN]をキーインすると、メニュー画面に戻る。

●登録の変更

データの内容を変更するときは、メニュー画面で[2]をキーイン。変更の画面になる。この処理は、氏名や電話番号の変更のほかに、データをすべて消して新しい人のデータを入れるときにも使う。

手順は、次のとおり。

①変更したいデータの氏名番号をキーイン。しばらくして、そのデータと変更するかどうかのメッセージがあらわれる。変更するなら、[Y][RETURN]。



④変更の画面(002をキーイン)



⑤氏名の変更



⑥電話番号の変更



⑦変更画面の入力終了

②カーソルが氏名のデータの先頭に移る。氏名を変更しない場合は、そのままで RETURN を押せばよい。

③カーソルが電話番号のデータの先頭に移る。変更の場合は新たに入力して、変更しない場合はそのままで RETURN 。

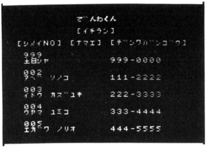
④変更がすべて終わると、登録するかどうかのメッセージがあらわれる。変更内容を確認して、Y RETURN 。

⑤新しい変更画面になるから、続けて変更するなら、①〜④の手順をくり返す。変更を終わる場合は、氏名番号の入力のところで 0 RETURN をキーインすると、メニュー画面に戻る。

◉一覧の表示

新規または変更処理で登録されたデータは、ファイルとしてディスクの中に収納されている（電源を切っても消えない）。メニュー画面で 3 をキーインすると、ファイルのデータを一覧表示で見ることができる。

表示される順序は、氏名番号や登録の順序に関係なく、50音順。正確にはキャラクタコード（マニュアル参照）で判断されるから、グラフィック・シンボルの漢字→アルファベット大文字→小文字→ひらがな（あ〜そ）→カタカナ→ひらが



⑧一覧画面——最初の５件



⑨次の５件

な（た〜ん）の順で並びかえられる。

一覧で一度に表示されるのは、5件ぶん。RETURN キーを押すと次の5件、もう一度押すとその次の5件、というぐあいに表示されていく。

登録されたデータがすべて表示された段階で RETURN キー（この場合は、ほかのどのキーでもよい）を押すと、メニュー画面に戻る。

◉終了

このゲームを終了するには、どの処理でもいったんメニュー画面に戻し、9をキーインする。プログラムの実行が終わり、次の命令待ちとなる。

前述したように、ファイルは電源を切っても保存されているから、次にこのプログラムを実行すると、最初のメニュー画面で登録数が示される。

ゲームのコツと注意

◎登録、変更でYかNをキーインするときは、大文字しか受けつけない。CAPS LOCK キーをロックし、かな キーをロックしてある場合は解除しておこう。

◎いずれの場合も、キーインをまちがった場合はブザーが鳴るから、もう一度、正確に入力しなおせばよい。

◎登録できるのは、氏名番号が3桁、氏名と電話番号が12桁まで。それ以上入力しても、画面では表示されるが、内部の記憶からは消えてしまう。

◎データの削除はできないから、変更処理を利用して、同じ氏名番号の内容をそっくり入れかえるしかない。

◎氏名の入力は、一覧表示で見やすくするために、特殊な場合を除いてカタカナかひらがなで統一したほうがよい。

◎会社名、店名など特殊なグループは、(K)、#、&などの記号を頭につけておくと、一覧でまとめて表示されるから使いやすい。

プログラムの解説

▶初期設定と処理の流れ〈行100〜160〉

行100 配列の領域を確保するために、メモリ領域の大きさを指定する。

行110 すべての変数を整数型に宣言する。

行120 ファンクション・キーの表示を消す。

行130 配列変数を宣言する。50件ぶんのデータが入る。

行140 メニュー画面の処理を呼び出して、表示する。

行150 変数PT（処理の選択）の値によって、登録、変更、一覧のそれぞれの処理を呼び出し、実行する。

行160 メニュー画面の表示処理に戻る。

▶メニュー画面の表示〈行170〜310〉

行180〜230 処理選択の説明を表示する。

行240 一覧処理から戻ったときは、以下の処理を行わず、行280へ飛ぶ。

行250〜270 行250の表示のあと、ファイルにデータが何件入っているかをかぞえ、行270でその表示をする。

行280〜290 キーインを待ち、その内容を数値に変える。

行300 9がキーインされたとき、プログラムの実行を終わる。

行310 1〜3以外のキーが押されたときは、ブザーを鳴らし、再びキーインを待つ。

▶登録処理〈行320〜460〉

行330〜340 画面のタイトルを表示する。

行350 画面の表示処理を呼び出して、実行する。

行360 氏名番号の入力処理を実行する。0がキーインされたときは、メニュー画面に戻る。

行370 ファイルのデータを調べ、登録ずみかどうかをチェックする。

行380 登録ずみの場合はそのメッセージを出し、ブザーを3回鳴らして行360に戻る。

行390〜400 氏名、電話番号の入力処理を、それぞれ呼び出して実行する。

行410〜460 登録するかどうかの表示を出し、キーインを待つ。[Y]がキーインされたら、ファイルにデータを書きこみ、行330に戻って次の登録画面に移る。[N]がキーインされたら、画面を消して登録画面を元に戻す。それ以外のときは、ブザ

ーを鳴らし、再びキーインを待つ。

▶変更処理〈行470～660〉

行480～490　画面のタイトルを表示する。

行500　画面の表示処理を呼び出して、実行する。

行510　氏名番号の入力処理を実行する。0がキーインされたときは、メニュー画面に戻る。

行520～530　ファイルに登録されているかどうかを調べ、登録されていなければそのメッセージを出し、ブザーを3回鳴らして行510に戻る。

行540　登録されていれば、そのデータを画面に表示する。

行550～600　変更するかどうかのメッセージを出し、Yがキーインされたら、氏名、電話番号の入力処理をして、行610へ飛ぶ。Nがキーインされたら、画面を消して、変更画面を元に戻す。それ以外のときは、ブザーを鳴らし、再びキーインを待つ。

行610～660　変更内容を登録するかどうかのメッセージを出し、以下、行410～460と同様の処理をする。

▶一覧処理〈行670～740〉

行680～710　一覧画面の表示。

行720　一覧処理を実行したかどうかの信号になる変数DFに1をセットし、ファイルのデータ数をかぞえる。データ数が0のときは、ブザーを2回鳴らしてメニュー画面に戻る。

行730　登録されたデータが1件のときは、そのデータを読みこみ、一覧画面に表示する。

行740　データ数が2以上のときは、データを読みこみ、それをキャラクタコードの順に並べかえて表示する。

▶データの入力と変更〈行750～960〉

行750～780　氏名番号（3桁）の入力を待って、それを文字型変数SNO$に入れる。

行790～870　氏名（12桁）の入力を待って、それを文字型変数NA$に入れる。

行880～960　電話番号（12桁）の入力を待って、それを文字型変数TEL$に入れる。

▶画面の表示〈行970～1080〉

行980～1080　データ入力の画面をつくり、表示する。

▶データの表示〈行1090〜1120〉

行1100〜1120　入力された氏名と電話番号を表示する（変更処理用）。

▶データの呼び出し、書きこみとカウント〈行1130〜1370〉

行1140　ファイル「TELTEL．DAT」を開く。

行1150〜1180　ファイルの中の氏名、電話番号、氏名番号をそれぞれ12、12、3桁に設定し、データが何件めかを示す変数Rを用意する。行1180でRに1プラスする。

行1190　入力された氏名番号と同じ番号がないときは、NFを1にして呼び出し処理を終わる。同じ番号があるときは、NFを0にしておく。

行1200　R番めの氏名番号を読み、入力されたものと違う場合は行1180に戻り、R＋1番めを読む。

行1210　ファイルのR番めの氏名番号と入力された氏名番号が同じ場合は、R番めのデータを読み、変数に入れる。

行1220〜1230　ファイルを閉じ、データの呼び出し処理を終わる。

行1250〜1260　ファイルを開き、氏名、電話番号、氏名番号をそれぞれ12、12、3桁に設定する。

行1270〜1300　それぞれの内容を左詰めで取りこみ、データ（レコード）数をかぞえてファイルに書きこむ。たとえば氏名番号なら、図のように書きこまれる。また、書きこむ場所CTは、行1290によって**PT**＝2(変更)なら元のデータと同じ場所（R番め）に、それ以外なら次のレコード（C＋1番め）になる。



行1310〜1320　ファイルを閉じて、書きこみ処理を終わる。

行1330〜1370　ファイルのデータ（レコード）数をかぞえる処理。ファイルを開き、レコード数をかぞえ（1レコードは256バイト）、ファイルを閉じて処理を終わる。

▶一覧画面の表示処理〈行1380〜1530〉

行1390　「シバラク　オマチクダサイ」の表示を消す。

行1400〜1460　1画面に5件ぶんのデータを表示するループ。

行1470　登録ずみのデータをすべて表示し終わったら、行1530へ飛ぶ。

行1480〜1490　何かキーが押されるのを待ち、RETURN キーが押されたら、次の

処理に移る。

行1500〜1520　画面のデータを消し、次のデータ表示（5件ぶん）に移るループ。

行1530　何かキーが押されるのを待ち、押されたらメニュー画面に戻る。

▶データの読みこみと並べかえ 〈行1540〜1700〉

行1550〜1570　ファイルを開き、氏名、電話番号、氏名番号の桁(けた)数を設定してデータ数をかぞえる。

行1580〜1610　並べかえのため、R番めのデータをそれぞれの配列変数に入れるループ。

行1620〜1630　ファイルを閉じて、データの読みこみ処理を終わる。

行1640〜1700　データ（氏名番号、氏名、電話番号）を、氏名のキャラクタコードの順に並べかえる処理。

使用変数

C　ランダムファイルに登録されたレコード件数

DF　一覧表示処理をしたときに、それを示す値として1をセットする

**NA$**　氏名のデータを入れる。ファイルのZNA$に対応

NF　ファイルに登録ずみ(＝0)か、そうでない(＝1)かを示す

PT　メニュー画面で処理の選択番号

R　ファイルの何番めかということを示す。CTも同じ

**SNO$**　氏名番号のデータを入れる。ファイルのZSNO$に対応

**TEL$**　電話番号のデータを入れる。ファイルのZTEL$に対応



```
99 '  TELﾌｧｲﾙ
100 CLEAR 10000!
110 DEFINT A-Z
120 KEY OFF
130 DIM SNO$(50),NA$(50),TEL$(50)
140 GOSUB 180
150 ON PT GOSUB 330,480,680
160 GOTO 140
170 '--- ﾒﾆｭｰ ---
180 CLS:LOCATE 10,2:PRINT"でﾞんわくん"
190 LOCATE 10,5:PRINT"[ﾒﾆｭｰ]"
200 LOCATE 6,8:PRINT"1. ﾄｳﾛｸ"
210 LOCATE 16,8:PRINT"2. ﾍﾝｺｳ"
220 LOCATE 6,11:PRINT"3. ｲﾁﾗﾝ"
230 LOCATE 16,11:PRINT"9. ｵﾜﾘ"
240 IF DF=1 THEN GOTO 280
250 LOCATE 6,17:PRINT"ｼﾊﾞﾗｸ ｵﾏﾁｸﾀﾞｻｲ"
260 GOSUB 1340
270 LOCATE 2,17:PRINT"<ｹﾞﾝｻﾞｲ";C;"ﾒｲ ﾄｳﾛｸｻﾚﾃｲﾏｽ>"
280 LOCATE 4,20:PRINT"ﾊﾞﾝｺﾞｳ ｦ ｴﾗﾝﾃﾞｸﾀﾞｻｲ"
290 A$=INKEY$:IF A$="" THEN 290 ELSE PT=VAL(A$)
300 IF PT=9 THEN CLS:END
310 IF PT<1 OR PT>3 THEN BEEP:GOTO 290 ELSE RETURN
320 '--- ﾄｳﾛｸ ---
330 CLS:LOCATE 10,2:PRINT"でﾞんわくん"
340 LOCATE 10,4:PRINT"[ﾄｳﾛｸ]"
350 GOSUB 980
360 GOSUB 760:IF SNO$="0" THEN RETURN
370 GOSUB 1140
380 IF NF=0 THEN LOCATE 4,20:PRINT"ｽﾃﾞﾆ ﾄｳﾛｸｽﾞﾐﾃﾞｽ!":BEEP:BEEP:BEEP:LOCATE 4,20:
PRINT SPC(20):GOTO 360
390 GOSUB 800
400 GOSUB 890
```

```
410 LOCATE 4,20:PRINT"ﾄｳﾛｸ ｼﾏｽｶ? <Y/N>"
420 LOCATE 22,21:PRINT"-"
430 LOCATE 22,20:LINE INPUT IN$
440 IF IN$="Y" THEN GOSUB 1250:GOTO 330
450 IF IN$="N" THEN GOTO 330
460 BEEP:GOTO 430
470 '--- ﾍﾝｺｳ ---
480 CLS:LOCATE 10,2:PRINT"て゛んわくん"
490 LOCATE 10,4:PRINT"[ﾍﾝｺｳ]"
500 GOSUB 980
510 GOSUB 760:IF SNO$="0" THEN RETURN
520 GOSUB 1140
530 IF NF=1 THEN LOCATE 4,20:PRINT"ﾄｳﾛｸ ｻﾚﾃ ｲﾏｾﾝ!":BEEP:BEEP:BEEP:LOCATE 4,20:PR
INT SPC(20):GOTO 510
540 GOSUB 1100
550 LOCATE 4,20:PRINT"ﾍﾝｺｳ ｼﾏｽｶ? <Y/N>"
560 LOCATE 22,21:PRINT"-"
570 LOCATE 22,20:LINE INPUT IN$
580 IF IN$="Y" THEN GOSUB 840:GOSUB 930:GOTO 610
590 IF IN$="N" THEN GOTO 480
600 BEEP:GOTO 570
610 LOCATE 4,20:PRINT"ﾄｳﾛｸ ｼﾏｽｶ? <Y/N>"
620 LOCATE 22,21:PRINT" "
630 LOCATE 22,20:PRINT SPC(1):LOCATE 22,20:LINE INPUT IN$
640 IF IN$="Y" THEN GOSUB 1250:GOTO 480
650 IF IN$="N" THEN GOTO 480
660 BEEP:GOTO 630
670 '--- ｲﾁﾗﾝ ---
680 CLS:LOCATE 10,2:PRINT"て゛んわくん"
690 LOCATE 10,4:PRINT"[ｲﾁﾗﾝ]
700 LOCATE 1,6:PRINT"[ｼﾒｲNO] [ﾅﾏｴ] [ﾃﾞﾝﾜﾊﾞﾝｺﾞｳ]"
710 LOCATE 6,17:PRINT"ｼﾊﾞﾗｸ ｵﾏﾁｸﾀﾞｻｲ"
720 DF=1:GOSUB 1340:IF C=0 THEN BEEP:BEEP:RETURN
```

```
730 IF C=1 THEN GOSUB 1550:GOSUB 1390:RETURN
740 GOSUB 1550:GOSUB 1650:GOSUB 1390:RETURN
750 '--- ｼﾒｲNO ﾉ ﾆｭｳﾘｮｸ ---
760 LOCATE 12,8:PRINT SPC(3):LOCATE 12,8
770 LINE INPUT SNO$:IF SNO$="" THEN BEEP:GOTO 760
780 RETURN
790 '--- ｼﾒｲ ﾉ ﾆｭｳﾘｮｸ ---
800 LOCATE 12,12:PRINT SPC(12):LOCATE 12,12
810 LINE INPUT NA$:IF NA$="" THEN BEEP:GOTO 800
820 RETURN
830 '--- ｼﾒｲ ﾉ ﾍﾝｺｳ ---
840 LOCATE 12,12
850 A$=INKEY$:IF A$=CHR$(13) THEN NA$=ZNA$:GOTO 870
860 LINE INPUT NA$
870 RETURN
880 '--- ﾃﾞﾝﾜﾊﾞﾝｺﾞｳ ﾉ ﾆｭｳﾘｮｸ ---
890 LOCATE 12,16:PRINT SPC(12):LOCATE 12,16
900 LINE INPUT TEL$:IF TEL$="" THEN GOTO 890
910 RETURN
920 '--- ﾃﾞﾝﾜﾊﾞﾝｺﾞｳ ﾉ ﾍﾝｺｳ ---
930 LOCATE 12,16
940 A$=INKEY$:IF A$=CHR$(13) THEN TEL$=ZTEL$:GOTO 960
950 LINE INPUT TEL$
960 RETURN
970 '--- ｶﾞﾒﾝ ﾉ ﾋｮｳｼﾞ ---
980 LOCATE 4,8:PRINT"ｼﾒｲNO"
990 LOCATE 4,10:PRINT"<0> ﾃﾞ ﾒﾆｭｰｶﾞﾒﾝ ﾆ ﾓﾄﾞﾘﾏｽ"
1000 LOCATE 4,12:PRINT"ﾅﾏｴ"
1010 LOCATE 4,16:PRINT"TELNO"
1020 LOCATE 2,8:PRINT"♥"
1030 LOCATE 2,12:PRINT"♣"
1040 LOCATE 2,16:PRINT"♠"
1050 LOCATE 12,9:PRINT"---"
```

```
1060 LOCATE 12,13:PRINT"------------"
1070 LOCATE 12,17:PRINT"------------"
1080 RETURN
1090 '--- ﾃﾞｰﾀ ﾉ ﾋｮｳｼﾞ ---
1100 LOCATE 12,12:PRINT NA$
1110 LOCATE 12,16:PRINT TEL$
1120 RETURN
1130 '--- ﾃﾞｰﾀ ﾉ ﾖﾋﾞﾀﾞｼ ---
1140 OPEN "TELTEL.DAT" AS #1
1150 FIELD #1,12 AS ZNA$,12 AS ZTEL$,3 AS ZSNO$
1160 C=LOF(1)/256
1170 R=0
1180 R=R+1
1190 IF C<R THEN NF=1:CLOSE #1:RETURN ELSE NF=0
1200 GET #1,R:IF VAL(ZSNO$)<>VAL(SNO$) THEN GOTO 1180
1210 NA$=ZNA$:TEL$=ZTEL$:SNO$=ZSNO$
1220 CLOSE #1
1230 RETURN
1240 '--- ﾃﾞｰﾀ ﾉ ｶｷｺﾐ ---
1250 OPEN "TELTEL.DAT" AS #1
1260 FIELD #1,12 AS ZNA$,12 AS ZTEL$,3 AS ZSNO$
1270 LSET ZNA$=NA$:LSET ZTEL$=TEL$:LSET ZSNO$=SNO$
1280 C=LOF(1)/256
1290 IF PT=2 THEN CT=R ELSE CT=C+1
1300 PUT #1,CT
1310 CLOSE #1
1320 RETURN
1330 '--- ﾃﾞｰﾀ ﾉ ｶｳﾝﾄ ---
1340 OPEN "TELTEL.DAT" AS #1
1350 C=LOF(1)/256
1360 CLOSE #1
1370 RETURN
1380 '--- ｲﾁﾗﾝﾋｮｳ ﾉ ｶﾞﾒﾝ ---
```

```
1390 LOCATE 6,17:PRINT SPACE$(20)
1400 FOR I=8 TO 20 STEP 3
1410   M=M+1
1420   LOCATE 2,I:PRINT;SNO$(M)
1430   LOCATE 2,I+1:PRINT;NA$(M)
1440   LOCATE 15,I+1:PRINT TEL$(M)
1450   IF M=C THEN I=21
1460 NEXT I
1470 IF M=C THEN C=0:M=0:GOTO 1530
1480 GG$=INKEY$:IF GG$="" THEN 1480
1490 IF GG$<>CHR$(13) THEN BEEP:GOTO 1480
1500 FOR J=8 TO 21
1510   LOCATE 1,J:PRINT SPACE$(30)
1520 NEXT J:GOTO 1400
1530 GG$=INKEY$:IF GG$="" THEN 1530 ELSE RETURN
1540 '---ﾃﾞｰﾀ ﾉ ﾖﾐｺﾐ ---
1550 OPEN "TELTEL.DAT" AS #1
1560 FIELD #1,12 AS ZNA$,12 AS ZTEL$,3 AS ZSNO$
1570 C=LOF(1)/256
1580 FOR R=1 TO C
1590   GET #1,R
1600   SNO$(R)=ZSNO$:NA$(R)=ZNA$:TEL$(R)=ZTEL$
1610 NEXT R
1620 CLOSE #1
1630 RETURN
1640 '---ﾃﾞｰﾀ ﾉ ﾅﾗﾍﾞｶｴ ---
1650 FOR I=1 TO C-1
1660   FOR J= I+1 TO C
1670     IF NA$(I)>NA$(J) THEN W$=NA$(I):NA$(I)=NA$(J):NA$(J)=W$:U$=SNO$(I):SNO$
(I)=SNO$(J):SNO$(J)=U$:V$=TEL$(I):TEL$(I)=TEL$(J):TEL$(J)=V$
1680   NEXT J
1690 NEXT I
1700 RETURN
```

ふろく：1

エラーメッセージ一覧

（注） 比較的よく出てくるものだけを選んであります。

メッセージ	意　　味
Bad FAT	ディスクのフォーマットが正しく行われていない
Bad file mode	①シーケンシャル・ファイルに対して PUT 文、GET 文、LOF 関数を使った ②ランダム・ファイルに対して、LOAD 文が実行された ③OPEN 文の、モードの指定が不適当
Bad file number	①OPEN されていないファイル番号を指定した ②指定したファイル番号が、MAXFILES 文で設定した数を超えている
Bad sector number	PUT 文、GET 文で指定したレコード番号が 0 であるか、または32767より大きい
Disk full	ディスクがいっぱいで、書きこみできない
Disk I/O error	ディスクに傷などがついて、入出力できない
Disk offline	ディスクが装置にセットされていない
Disk write protected	ディスクが書きこみ禁止状態になっている
Division by zero	①0による割り算を行おうとした ②0に対して負のべき乗（0^{-5}など）を行った
FIELD overflow	FIELD 文で指定した領域が256バイトを超えた
File not found	指定したファイルがない
File not OPEN	指定したファイルが OPEN されていない
File still open	ファイルが CLOSE されていない
Illegal function call	①命令や関数の使い方が違う ②指定した数値などが、使用可能な範囲を超えている

メッセージ	意　　味
Input past end	ファイルをすべて読んだあと、さらに読もうとした
Missing operand	必要な指定が抜けている
NEXT without FOR	NEXT 文に対応する FOR 文がない
Out of DATA	READ で読みこませるはずのデータがない
Out of memory	プログラムが長すぎたり、配列が大きすぎたりして、メモリが足らなくなった
Out of string space	文字列変数用のメモリが足らなくなった
Overflow	演算結果や入力された数値が、型に応じた許容範囲を超えた
Redimensioned array	①配列変数を二重に定義しようとした ②配列宣言をしないで配列変数を使ったあと、再び同じ配列を宣言しようとした
RETURN without GOSUB	GOSUB 文と RETURN 文が、正しく対応していない
Sequential I/O only	シーケンシャル・ファイルに対して、GET 文、PUT 文を実行しようとした
Subscript out of range	①指定した配列変数の添え字の値が、DIM 文で宣言した値より大きい ②DIM 文で定義しないで使用した配列が10を超えた
Syntax error	文の書き方が違っている
Type mismatch	データの型が一致していない
Undefined line number	指定された行番号が、プログラム中にない
Undefined user function	DEF FN 文で定義されていない関数を使おうとした

ふろく：2

MSXベーシックの解説

(注) コマンド、ステートメント、関数、論理演算子、算術演算子のうちから、主なものだけを選んであります。解説のあとに（＊1）をつけたものはMSX 2に追加された命令、（＊2）をつけたものはディスク・ベーシックの命令です。

【A】

アブソリュート
ABS 数式の絶対値を求める関数。

アンド
AND 条件が二つあり、両方とも満たさなければならない場合、かつ、そして、という意味で使用する論理演算子。

アスキー
ASC 文字式のキャラクタコードを求める関数。

オート
AUTO 行番号を自動的に発生させる。

【B】

ビープ
BEEP ブザー音を鳴らす。

ビーロード
BLOAD 機械語のプログラムをメモリ上に読みこむ（＊2）。

ビーセーブ
BSAVE 機械語のプログラムを保存する（＊2）。

【C】

キャラクタ・ダラー
CHR$ 数式のキャラクタコードに対応する文字を与える関数。

サークル
CIRCLE グラフィック画面に、円やだ円を描く。

クリア
CLEAR 変数の初期化と、使用するメモリ領域の大きさを指定する。

シーロード
CLOAD カセットテープのプログラムを、メモリ上に読みこむ。

シーロード・ベリファイ
CLOAD? カセットテープ上のプログラムと、メモリ上のプログラムとを照合する。

クローズ
CLOSE ファイルを閉じる（＊2）。

クリアスクリーン
CLS 画面に表示されている文字や絵を消す。

カラー
COLOR 画面に表示する前景色、背景色、周辺色をその順で指定する。

カラー・スプライト
COLOR SPRITE スプライト・パターンの1列ごとの色を指定する（＊1）。

コンティニュー
CONT 停止したプログラムの実行を再開する。

コサイン
COS 数式のコサイン値を計算する関数。

シーセーブ
CSAVE メモリ上のプログラムをカセットテープに保存する。

コンバート・ダブル
CVD 文字列を、倍精度型の数値に変換する関数（＊2）。

コンバート・インテジャー
CVI 文字列を、整数型の数値に変換する関数（＊2）。

コンバート・シングル
CVS 文字列を、単精度型の数値に変換する関数（＊2）。

【D】

データ
DATA READ文で読みこむデータを用意する。

デファイン・ファンクション
DEF FN 関数を定義する。

デファイン・インテジャー/ストリング
DEF INT/STR　指定する文字で始まる変数を、整数型/文字型に宣言する。
デリート
DELETE　指定されたプログラム行を削除する。
ディメンジョン
DIM　配列変数を使うことを宣言する。
ドロー
DRAW　グラフィック画面に図形を描く。

【E】
エンド
END　プログラムの実行を終了する。
エンド・オブ・ファイル
EOF　ファイルが終了したかどうかを調べる関数（＊2）。
イレーズ
ERASE　配列変数を消去する。

【F】
フィールド
FIELD　ランダム入出力用のバッファ内の区切りを設定する（＊2）。
ファイルズ
FILES　フロッピィディスクに入っているファイル名を、画面に表示する（＊2）。
フォー・ネクスト
FOR～NEXT　FOR 文と NEXT 文にはさまれた命令を、指定された回数だけくり返す。

【G】
ゲット
GET　①文字変数に、現在の日付または時刻を代入する（＊1）。②ランダム・ファイルからバッファにデータを読みこむ（＊2）。
ゴーサブ
GOSUB　指定したサブルーチンを実行する。
ゴーツー
GOTO　指定した行番号に飛び、その行から実行する。

【H】
ヘキサ・ダラー
HEX$　数値を16進数の文字列に変換する関数。

【I】
イフ・ゼン・エルス
IF～THEN～ELSE　条件を判断し、条件を満たしていれば THEN 以下の処理を、満たしていなければ ELSE 以下の処理を実行する。THEN のかわりに GOTO を使い、IF～GOTO～ELSE と書くこともある。
インキー・ダラー
INKEY$　押されているキーの文字を求める関数。何も押されていなければ、ヌルストリング（何もない状態）。
インプット
INPUT　キーボードから入力されたデータを、指定した変数に入れる。
インプット・シャープ
INPUT#　ファイルからデータを読みこみ、変数に入れる（＊2）。
インプット・ダラー
INPUT$　指定されたファイルから、指定された長さの文字列を読み出す関数（＊2）。
インストリング
INSTR　文字列の中から指定された文字列を探し、その位置を返す関数。
インテジャー
INT　指定された値を超えない最大の整数を求める関数。
インターバル・オン/オフ/ストップ
INTERVAL ON/OFF/STOP　タイマーの割りこみ命令（ON INTERVAL GOSUB 命令）を、許可/禁止/一時中止する。

【K】
キー
KEY　ファンクション・キーの内容を定義する。
キー・リスト
KEY LIST　ファンクション・キーの

内容を、画面に表示する。

キー・オン/オフ
KEY ON/OFF　ファンクション・キーの内容を、画面に表示する/しない。

キル
KILL　フロッピィディスク内の指定したファイルを削除する（＊2）。

【L】

レフト・ダラー
LEFT$　文字式の左側から、指定された長さの文字列を取り出す関数。

レングス
LEN　文字式で指定された文字列の長さを求める関数。

レット
LET　変数に数値を代入する。LETは、省略してもよい。

ライン
LINE　グラフィック画面に、直線や四角形を描く。

ライン・インプット
LINE INPUT　キーボードから入力される1行全体の文字列を、指定した文字変数に入れる。

ライン・インプット・シャープ
LINE INPUT#　指定されたファイルから1行全体の文字列を読みこみ、文字変数に入れる（＊2）。

リスト
LIST　メモリにあるプログラムを、画面に表示する。

エルリスト
LLIST　メモリにあるプログラムを、プリンタに印刷する。

ロード
LOAD　フロッピィディスク内に保存してあるプログラムを、本体のメモリに読みこむ（＊2）。

ロケート
LOCATE　テキスト画面で、カーソルの位置を指定する。

ラスト・オブ・ファイル
LOF　ランダム・ファイルの最終レコード番号をかぞえる（＊2）。

レフト・セット
LSET　ランダム入出力用のバッファに、データを左詰めで書きこむ（＊2）。

【M】

マックスファイルズ
MAXFILES　使用するファイルの最大数を指定する（＊2）。

マージ
MERGE　メモリ上のプログラムと、フロッピィディスク内にセーブされているプログラムを、合わせて一つにする（＊2）。

ミドル・ダラー
MID$　①文字列の中から、指定した長さの文字列を取り出す関数。②文字変数の一部を、ほかの文字列に置きかえる命令。

メイク・ダブル・ダラー
MKD$　倍精度型の数値を、文字列に変換する関数（＊2）。

メイク・インテジャー・ダラー
MKI$　整数型の数値を、文字列に変換する関数（＊2）。

メイク・シングル・ダラー
MKS$　単精度型の数値を、文字列に変換する関数（＊2）。

モッド
MOD　割り算の、余りを求める算術演算子。

モーター・オン/オフ
MOTOR ON/OFF　カセットテープ・レコーダを動かす/停止させる。

【N】

ニュー
NEW　メモリ内のプログラムとデータを消去する。

ノット
NOT　～ではないという、否定の意味で使用する論理演算子。

【O】

オン・ゴーサブ
ON～GOSUB　条件によって、指定されたサブルーチンを実行する。

オン・ゴーツー
ON～GOTO　条件によって、指定された行番号へ飛ぶ。

オン・インターバル・ゴーサブ
ON INTERVAL GOSUB　指定した時間ごとにタイマー割りこみが起こるよう条件を設定し、割りこみ処理の開始行を指定する。

オン・キー・ゴーサブ
ON KEY GOSUB　ファンクション・キーが押されたときに呼び出す行を指定する。

オン・スプライト・ゴーサブ
ON SPRITE GOSUB　スプライト・パターンが衝突したときに呼び出す行を指定する。

オン・スティックトリガ・ゴーサブ
ON STRIG GOSUB　ジョイスティックのトリガボタンが押されたときに呼び出す行を指定する。

オープン
OPEN　フロッピィディスク内の指定したファイルを開く（＊２）。

オア
OR　条件が二つあって、どちらかを満たせばよいという場合、あるいはという意味で使用する論理演算子。

【P】

ペイント
PAINT　グラフィック画面の、指定した境界色で囲まれた部分を、指定した色でぬりつぶす。

プレイ
PLAY　①音楽を演奏する命令。②音楽を演奏中かどうかを調べる関数。

ポイント
POINT　グラフィック画面上の、指定した点の色を調べる関数。

ポイントリセット
PRESET　指定した点の色を変更する。背景色と同じ色を指定すれば、消すことになる。

プリント
PRINT　テキスト画面に、数値や文字列を表示する。

プリント・ユージング
PRINT USING　文字列や数値を、指定した書式でテキスト画面に表示する。

プリント・シャープ
PRINT #　指定したファイルに、データを書き出す（＊２）。

プリント・シャープ・ユージング
PRINT # USING　指定したファイルに、指定した書式でデータを書き出す（＊２）。

ポイントセット
PSET　グラフィック画面で、指定した位置に点を描く。

プット
PUT　ランダム入出力用のバッファ内のデータを、ファイルに書き出す（＊２）。

プット・スプライト
PUT SPRITE　スプライト・パターンを、画面に表示する。

プット・カンジ
PUT KANJI　SCREEN ５～８に漢字を表示する（＊１）。

【R】

リード
READ　DATE文で指定した数値や文字のデータを読みこみ、変数に入れる。

リマーク
REM　プログラム中に、注釈文を入れる。アポストロフィ（'）で代用することもできる。

リナンバ
RENUM　プログラムの行番号をつけなおす。

リストア
RESTORE　READ文で読みこむ先のDATA文を指定する。

リターン
RETURN　サブルーチンによる処理を終え、メイン処理に復帰する。

ライト・ダラー
RIGHT$　文字列の右側から、指定し

た長さの文字列を取り出す関数。

ランダム
RND　0以上、1未満の乱数を発生させる関数。

ライト・セット
RSET　ランダム入出力用のバッファに、データを右詰めで書きこむ(＊2)。

ラン
RUN　メモリにあるプログラムを実行する。

【S】

セーブ
SAVE　メモリ上のプログラムを、フロッピィディスク内に保存する(＊2)。

スクリーン
SCREEN　使用する画面、スプライトの大きさなどを設定する。

サイン
SGN　数式の符号を調べ、正であれば1、0であれば0、負であれば－1を出す関数。

サイン
SIN　数式のサイン値を計算する関数。

サウンド
SOUND　指定した番号に対応する音を出す。

スペース・ダラー
SPACE$　指定した長さの空白を求める関数。

スペース
SPC　画面やプリンタに、指定した長さの空白を表示する関数。

スプライト・ダラー
SPRITE$　スプライト・パターンを定義するシステム変数。

スプライト・オン/オフ/ストップ
SPRITE ON/OFF/STOP　スプライト・パターンが衝突したときの割りこみを、許可する/禁止する/一時中止する。

スクエアルート
SQR　数式のルート値を計算する関数。

スティック
STICK　カーソル移動キー(0)、またはジョイスティック（1～4）がどの方向に押されているかを求める関数。

ストリング・ダラー
STR$　数値を文字列に変える関数。

スティックトリガ
STRIG　ジョイスティックのトリガボタンが押されているかどうかを調べる関数。

スティックトリガ・オン/オフ/ストップ
STRIG(n) ON/OFF/STOP　スペース・キー（0)、またはジョイスティックのトリガボタン（1～4）が押されたときの割りこみを、許可する/禁止する/一時中止する。

ストリング・ダラー
STRING$　指定した文字を、指定した長さ（数）だけつくる関数。

スワップ
SWAP　二つの変数の値を交換する。

【T】

タブ
TAB　指定した桁(けた)まで空白を表示する関数。

タンジェント
TAN　数式のタンジェント値を計算する関数。

タイム
TIME　時間を計るシステム変数。

トレース・オン/オフ
TRON/TROFF　プログラムの実行状態を、追跡する/追跡をやめる。

TRONでは、実行した行番号が画面に表示される。

【V】

バリュー
VAL　文字列を数値に変換する関数。

【W】

ウイズ
WIDTH　画面の1行に表示する桁(けた)数を指定する。

●編著者紹介

日本ナレッジインダストリ株式会社（略称：JKI）

資本金＝9090万円、従業員＝300人、社長＝西尾　出

社名からもうかがえるように、同社は高度情報社会の担い手としてソフトウエア技術、情報通信技術、制御技術を駆使して産業システムや社会公共システムなどの研究開発に幅広い実績をあげている知識・情報産業会社である。

取り組む業務内容は実に多様であり、社会システムの調査、研究および企業の高度情報システムに関するコンサルティングには特に強い。またニューメディア関連システム、マイクロプロセッサ応用システムの開発、さらに医療システムや交通システムの研究開発、教育システムの研究開発、および関連付帯事業としての技術手引書・書籍の執筆・監修など幅広い活動にも積極的に取り組んでいる。こうした分野で仕事をしたい人々に対しては常時門戸を開いているというから、関心のある人はその門を叩いてみるとよい。

本書についてのお問い合わせは(ご質問内容を詳しくお書きください)、必ずハガキまたは手紙にて、弊社編集部あてにご郵送ください。

MSXたのしいゲームの作り方

編著者　日本ナレッジインダストリ株式会社　1987年6月20日発行

発行者　若　松　親　光

発行所　株式会社　西東社（せいとうしゃ）

〒101　東京都千代田区神田錦町3-15
☎東京（03）２９１－５８１５（代）
FAX　東京（03）２３３－２４６６（代）
振替口座番号／東京　８－６４９９

ISBN4-7916-0353-2

読者の皆さまへ

弊社の出版物は、全国有名小売書店の店頭に備えてあります。実物をごらんになって、お買い求めください。

書店に品切れの節は、店員にご下命になれば、弊社から取り寄せてお渡しいたします。

ぜひ、お近くの書店をご利用ください。

株式会社 西東社（せいとうしゃ）

東京都千代田区神田錦町３丁目15番地
電話　東京（03）291－５８１５(代表)